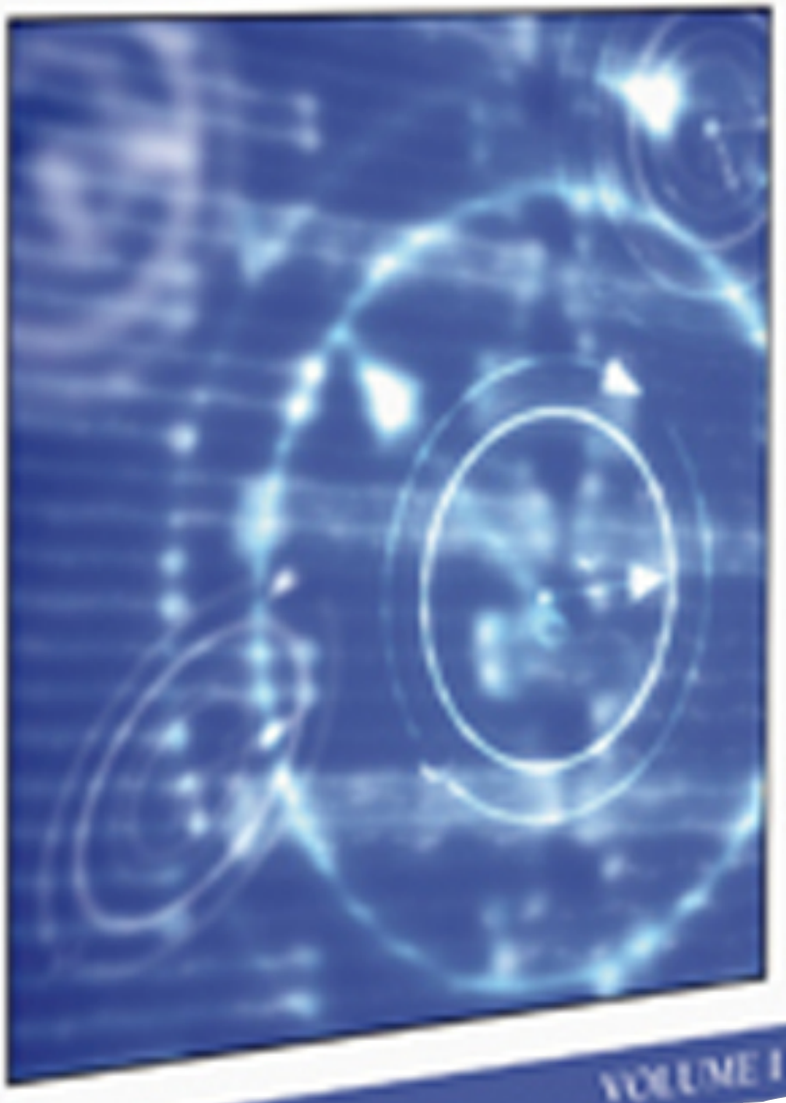
ENCYCLOPEDIA OF

DATA WAREHOUSING AND MINING

SECOND EDITION



JOHN WANG

VOLUME I

WANG

Encyclopedia of Data Warehousing and Mining - Second Edition

Volume II

REFERENCE

WANG

Encyclopedia of Data Warehousing and Mining - Second Edition

Volume III

REFERENCE

WANG

Encyclopedia of Data Warehousing and Mining - Second Edition

Volume IV

REFERENCE

Encyclopedia of Data Warehousing and Mining

Second Edition

John Wang
Montclair State University, USA

Information Science REFERENCE

INFORMATION SCIENCE REFERENCE
Hershey · New York

Director of Editorial Content: Kristin Klinger
Director of Production: Jennifer Neidig
Managing Editor: Jamie Snavely
Assistant Managing Editor: Carole Coulson
Typesetter: Amanda Appicello, Jeff Ash, Mike Brehem, Carole Coulson, Elizabeth Duke, Jen Henderson, Chris Hrobak, Jennifer Neidig, Jamie Snavely, Sean Woznicki
Cover Design: Lisa Tosheff
Printed at: Yurchak Printing Inc.

Published in the United States of America by
Information Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue, Suite 200
Hershey PA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: http://www.igi-global.com/reference

and in the United Kingdom by
Information Science Reference (an imprint of IGI Global)
3 Henrietta Street
Covent Garden
London WC2E 8LU
Tel: 44 20 7240 0856
Fax: 44 20 7379 0609
Web site: http://www.eurospanbookstore.com

Library of Congress Cataloging-in-Publication Data

Encyclopedia of data warehousing and mining / John Wang, editor. -- 2nd ed.
p. cm.
Includes bibliographical references and index.
Summary: "This set offers thorough examination of the issues of importance in the rapidly changing field of data warehousing and mining"--Provided by publisher.
ISBN 978-1-60566-010-3 (hardcover) -- ISBN 978-1-60566-011-0 (ebook)
1. Data mining. 2. Data warehousing. I. Wang, John,
QA76.9.D37E52 2008
005.74--dc22
2008030801

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

Editorial Advisory Board

List of Contributors

Contents
by Volume

VOLUME I

VOLUME II

VOLUME III

VOLUME IV

Contents
by Topic

Association

Bioinformatics

Classification

Clustering

Complex Data

Constraints

CRM

Data Cube & OLAP

Data Preparation

Data Streams

Data Warehouse

Decision

Decision Trees

Dempster-Shafer Theory

Dimensionality Reduction

Distributed Data Mining

Dynamic Data Mining

E-Mail Security

Ensemble Methods

Entity Relation

Evaluation

Evolutionary Algorithms

Explanation-Oriented

Facial Recognition

Feature

Fraud Detection

GIS

Government

Graphs

Health Monitoring

Image Retrieval

Information Fusion

Integration

Intelligence

Interactive

Internationalization

Kernal Methods

Knowledge

Large Datasets

Latent Structure

Manifold Alignment

Meta-Learning

Modeling

Multi-Agent Systems

Multimedia

Music

Negation

Neural Networks

News Recommendation

Ontologies

Optimization

Order Preserving

Outlier

Partitioning

Pattern

Privacy

Process Mining

Production

Program Comprehension

Program Mining

Provenance

Proximity

Receiver Operating Characteristics

Score Distribution Models

Search

Security

Segmentation

Self-Tuning Database

Sentiment Analysis

Sequence

Service

Shape Mining

Similarity/Disimilarity

Soft Computing

Software

Statistical Approaches

Symbiotic

Synthetic Databases

Text Mining

Time Series

Tool Selection

Unlabeled Data

Visualization

Web Mining

XML

Foreword

Since my foreword to the first edition of the Encyclopedia written over three years ago, the field of data mining continued to grow with more researchers coming to the field from a diverse set of disciplines, including statistics, machine learning, databases, mathematics, OR/management science, marketing, biology, physics and chemistry, and contributing to the field by providing different perspectives on data mining for an ever-growing set of topics.

This cross-fertilization of ideas and different perspectives assures that data mining remains a rapidly evolving field with new areas emerging and the old ones undergoing major transformations. For example, the topic of mining networked data witnessed very significant advances over the past few years, especially in the area of mining social networks and communities of practice. Similarly, text and Web mining undergone significant evolution over the past few years and several books and numerous papers have been recently published on these topics.

Therefore, it is important to take periodic "snapshots" of the field every few years, and this is the purpose of the second edition of the Encyclopedia. Moreover, it is important not only to update previously published articles, but also to provide fresh perspectives on them and other data mining topics in the form of new overviews of these areas. Therefore, the second edition of the Encyclopedia contains the mixture of the two—revised reviews from the first edition and new ones written specifically for the second edition. This helps the Encyclopedia to maintain the balanced mixture of the old and the new topics and perspectives.

Despite all the progress made in the data mining field over the past 10–15 years, the field faces several important challenges, as observed by Qiang Yang and Xindong Wu in their presentation "10 Challenging Problems in Data Mining Research" given at the IEEE ICDM Conference in December 2005 (a companion article was published in the *International Journal of Information Technology & Decision Making*, 5(4) in 2006). Therefore, interesting and challenging work lies ahead for the data mining community to address these and other challenges, and this edition of the Encyclopedia remains what it is—a milestone on a long road ahead.

Alexander Tuzhilin
New York
December 2007

Preface

How can a manager get out of a data-flooded "mire"? How can a confused decision maker navigate through a "maze"? How can an over-burdened problem solver clean up a "mess"? How can an exhausted scientist bypass a "myth"?

The answer to all of these is to employ a powerful tool known as data mining (DM). DM can turn data into dollars; transform information into intelligence; change patterns into profit; and convert relationships into resources.

As the *third* branch of operations research and management science (OR/MS) and the *third* milestone of data management, DM can help support the *third* category of decision making by elevating raw data into the *third* stage of knowledge creation.

The term "third" has been mentioned four times above. Let's go backward and look at three stages of knowledge creation. Managers are drowning in data (the first stage) yet starved for knowledge. A collection of data is not information (the second stage); yet a collection of information is not knowledge! Data are full of information which can yield useful knowledge. The whole subject of DM therefore has a synergy of its own and represents more than the sum of its parts.

There are three categories of decision making: structured, semi-structured and unstructured. Decision making processes fall along a continuum that range from highly structured decisions (sometimes called programmed) to much unstructured, non-programmed decision making (Turban et al., 2006).

At one end of the spectrum, structured processes are routine, often repetitive, problems for which standard solutions exist. Unfortunately, rather than being static, deterministic and simple, the majority of real world problems are dynamic, probabilistic, and complex. Many professional and personal problems can be classified as unstructured, semi-structured, or somewhere in between. In addition to developing normative models (such as linear programming, economic order quantity) for solving structured (or programmed) problems, operation researchers and management scientists have created many *descriptive* models, such as simulation and goal programming, to deal with semi-structured tasks. Unstructured problems, however, fall into a gray area for which there is no cut-and-dry solution. The current two branches of OR/MS often cannot solve unstructured problems effectively.

To obtain knowledge, one must understand the patterns that emerge from information. Patterns are not just simple relationships among data; they exist separately from information, as archetypes or standards to which emerging information can be compared so that one may draw inferences and take action. Over the last 40 years, the tools and techniques used to process data and information have continued to evolve from databases (DBs) to data warehousing (DW), to DM. DW applications, as a result, have become business-critical and can deliver even more value from these huge repositories of data.

Certainly, there are many statistical models that have emerged over time. Earlier, machine learning has marked a milestone in the evolution of computer science (Fayyad, Piatetsky-Shapiro, Smyth & Uthurusamy, 1996). Although DM is still in its infancy, it is now being used in a wide range of industries and for a range of tasks and contexts (Wang, 2006). DM is synonymous with knowledge discovery in databases, knowledge extraction, data/pattern analysis, data archeology, data dredging, data snooping, data fishing, information harvesting, and

business intelligence (Hand et al., 2001; Giudici, 2003; Han & Kamber, 2006). Data warehousing and mining (DWM) is the science of managing and analyzing large datasets and discovering novel patterns within them. In recent years, DWM has emerged as a particularly exciting and relevant area of research. Prodigious amounts of data are now being generated in domains as diverse and elusive as market research, functional genomics, and pharmaceuticals and intelligently analyzing them to discover knowledge is the challenge that lies ahead.

Yet managing this flood of data, and making it useful and available to decision makers has been a major organizational challenge. We are facing and witnessing global trends (e.g. an information/knowledge-based economy, globalization, technological advances etc.) that drive/motivate data mining and data warehousing research and practice. These developments pose huge challenges (eg. need for faster learning, performance efficiency/effectiveness, new knowledge and innovation) and demonstrate the importance and role of DWM in responding to and aiding this new economy through the use of technology and computing power. DWM allows the extraction of "nuggets" or "pearls" of knowledge from huge historical stores of data. It can help to predict outcomes of future situations, to optimize business decisions, to increase the customer relationship management, and to improve customer satisfaction. As such, DWM has become an indispensable technology for businesses and researchers in many fields.

The Encyclopedia of Data Warehousing and Mining (2nd Edition) provides theories, methodologies, functionalities, and applications to decision makers, problem solvers, and data mining professionals and researchers in business, academia, and government. Since DWM lies at the junction of database systems, artificial intelligence, machine learning and applied statistics, it has the potential to be a highly valuable area for researchers and practitioners. Together with a comprehensive overview, *The Encyclopedia of Data Warehousing and Mining* (2nd Edition) offers a thorough exposure to the issues of importance in this rapidly changing field. The encyclopedia also includes a rich mix of introductory and advanced topics while providing a comprehensive source of technical, functional, and legal references to DWM.

After spending more than two years preparing this volume, using a totally peer-reviewed process, I am pleased to see it published. Of the *324* articles, there are *214* brand-new articles and 110 updated ones that were chosen from the *234* manuscripts in the first edition. Clearly, the need to significantly update the encyclopedia is due to the tremendous progress in this ever-growing field. Our selection standards were very high. Each chapter was evaluated by at least three peer reviewers; additional third-party reviews were sought in cases of controversy. There have been numerous instances where this feedback has helped to improve the quality of the content, and guided authors on how they should approach their topics. The primary objective of this encyclopedia is to explore the myriad of issues regarding DWM. A broad spectrum of practitioners, managers, scientists, educators, and graduate students who teach, perform research, and/or implement these methods and concepts, can all benefit from this encyclopedia.

The encyclopedia contains a total of *324* articles, written by an international team of *555* experts including leading scientists and talented young scholars from *over forty* countries. They have contributed great effort to create a source of solid, practical information source, grounded by underlying theories that should become a resource for all people involved in this dynamic new field. Let's take a peek at a few articles:

Kamel presents an overview of the most important issues and considerations for preparing data for DM. Practical experience of DM has revealed that preparing data is the most time-consuming phase of any DM project. Estimates of the amount of time and resources spent on data preparation vary from at least 60% to upward of 80%. In spite of this fact, not enough attention is given to this important task, thus perpetuating the idea that the core of the DM effort is the modeling process rather than all phases of the DM life cycle.

The past decade has seen a steady increase in the number of fielded applications of predictive DM. The success of such applications depends heavily on the selection and combination of suitable pre-processing and modeling algorithms. Since the expertise necessary for this selection is seldom available in-house, users must either resort to trial-and-error or consultation of experts. Clearly, neither solution is completely satisfactory for the non-expert end-users who wish to access the technology more directly and cost-effectively. Automatic and systematic guidance is required. Giraud-Carrier, Brazdil, Soares, and Vilalta show how meta-learning can be leveraged to provide such guidance through effective exploitation of meta-knowledge acquired through experience.

Ruqian Lu has developed a methodology of acquiring knowledge automatically based on pseudo-natural language understanding. He has won two first class awards from the Academia Sinica and a National second class prize. He has also won the sixth Hua Loo-keng Mathematics Prize.

Wu, McGinnity, and Prasad present a general self-organizing computing network, which have been applied to a hybrid of numerical machine learning approaches and symbolic AI techniques to discover knowledge from databases with a diversity of data types. The authors have also studied various types of bio-inspired intelligent computational models and uncertainty reasoning theories. Based on the research results, the IFOMIND robot control system won the 2005 Fourth British Computer Society's Annual Prize for Progress towards Machine Intelligence.

Zhang, Xu, and Wang introduce a class of new data distortion techniques based on matrix decomposition. They pioneer use of *Singular Value Decomposition* and *Nonnegative Matrix Factorization* techniques for perturbing numerical data values in privacy-preserving DM. The major advantage of this class of data distortion techniques is that they perturb the data as an entire dataset, which is different from commonly used data perturbation techniques in statistics.

There are often situations with large amounts of "unlabeled data" (where only the explanatory variables are known, but the target variable is not known) and with small amounts of labeled data. As recent research in machine learning has shown, using only labeled data to build predictive models can potentially ignore useful information contained in the unlabeled data. Yang and Padmanabhan show how learning patterns from the entire data (labeled plus unlabeled) can be one effective way of exploiting the unlabeled data when building predictive models.

Pratihar explains the principles of some of the non-linear *Dimensionality Reduction* (DR) techniques, namely Sammon's *Non-Linear Mapping* (NLM), *VISOR algorithm*, *Self-Organizing Map* (SOM) and *Genetic Algorithm* (GA)-Like Technique. Their performances have been compared in terms of accuracy in mapping, visibility and computational complexity on a test function – Schaffer's F1. The author had proposed the above GA-like Technique, previously.

A lot of projected clustering algorithms that focus on finding specific projection for each cluster have been proposed very recently. Deng and Wu found in their study that, besides distance, the closeness of points in different dimensions also depends on the distributions of data along those dimensions. Based on this finding, they propose a projected clustering algorithm, IPROCLUS (Improved PROCLUS), which is efficient and accurate in handling data in high dimensional space. According to the experimental results on real biological data, their algorithm shows much better accuracy than PROCLUS.

Meisel and Mattfeld highlight and summarize the state of the art in attempts to gain synergies from integrating DM and Operations Research. They identify three basic ways of integrating the two paradigms as well as discuss and classify, according to the established framework, recent publications on the intersection of DM and Operations Research.

Yuksektepe and Turkay present a new data classification method based on *mixed-integer programming*. Traditional approaches that are based on partitioning the data sets into two groups perform poorly for multi-class data classification problems. The proposed approach is based on the use of hyper-boxes for defining boundaries of the classes that include all or some of the points in that set. A *mixed-integer programming* model is developed for representing existence of hyper-boxes and their boundaries.

Reddy and Rajaratnam give an overview of the *Expectation Maximization* (EM) algorithm, deriving its theoretical properties, and discussing some of the popularly used global optimization methods in the context of this algorithm. In addition the article provides details of using the EM algorithm in the context of the finite mixture models, as well as a comprehensive set of derivations in the context of *Gaussian* mixture models. Also, it shows some comparative results on the performance of the EM algorithm when used along with popular global optimization methods for obtaining maximum likelihood estimates and the future research trends in the EM literature.

Smirnov, Pashkin, Levashova, Kashevnik, and Shilov describe usage of an *ontology-based context model* for decision support purposes and document ongoing research in the area of intelligent decision support based on context-driven knowledge and information integration from distributed sources. Within the research the context is used for representation of a decision situation to the decision maker and for support of the decision maker in

solving tasks typical for the presented situation. The solutions and the final decision are stored in the user profile for further analysis via decision mining to improve the quality of the decision support process.

Corresponding to Feng, XML-enabled association rule framework extends the notion of associated items to XML fragments to present associations among trees rather than simple-structured items of atomic values. They are more flexible and powerful in representing both simple and complex structured association relationships inherent in XML data. Compared with traditional association mining in the well-structured world, mining from XML data, however, is confronted with more challenges due to the inherent flexibilities of XML in both structure and semantics. To make XML-enabled association rule mining truly practical and computationally tractable, template-guided mining of association rules from large XML data must be developed.

With the XML becoming a standard for representing business data, a new trend toward XML DW has been emerging for a couple of years, as well as efforts for extending the XQuery language with near-OLAP capabilities. Mahboubi, Hachicha, and Darmont present an overview of the major XML warehousing approaches, as well as the existing approaches for performing OLAP analyses over XML data. They also discuss the issues and future trends in this area and illustrate this topic by presenting the design of a unified, XML DW architecture and a set of XOLAP operators.

Due to the growing use of XML data for data storage and exchange, there is an imminent need for developing efficient algorithms to perform DM on semi-structured XML data. However, the complexity of its structure makes mining on XML much more complicated than mining on relational data. Ding discusses the problems and challenges in XML DM and provides an overview of various approaches to XML mining.

Pon, Cardenas, and Buttler address the unique challenges and issues involved in personalized online news recommendation, providing background on the shortfalls of existing news recommendation systems, traditional document adaptive filtering, as well as document classification, the need for online feature selection and efficient streaming document classification, and feature extraction algorithms. In light of these challenges, possible machine learning solutions are explored, including how existing techniques can be applied to some of the problems related to online news recommendation.

Clustering is a DM technique to group a set of data objects into classes of similar data objects. While Peer-to-Peer systems have emerged as a new technique for information sharing on Internet, the issues of peer-to-peer clustering have been considered only recently. Li and Lee discuss the main issues of peer-to-peer clustering and reviews representation models and communication models which are important in peer-to-peer clustering.

Users must often refine queries to improve search result relevancy. Query expansion approaches help users with this task by suggesting refinement terms or automatically modifying the user's query. Finding refinement terms involves mining a diverse range of data including page text, query text, user relevancy judgments, historical queries, and user interaction with the search results. The problem is that existing approaches often reduce relevancy by changing the meaning of the query, especially for the complex ones, which are the most likely to need refinement. Fortunately, the most recent research has begun to address complex queries by using semantic knowledge and Crabtree's paper provides information about the developments of this new research.

Li addresses web presence and evolution through web log analysis, a significant challenge faced by electronic business and electronic commerce given the rapid growth of the WWW and the intensified competition. Techniques are presented to evolve the web presence and to produce ultimately a predictive model such that the evolution of a given web site can be categorized under its particular context for strategic planning. The analysis of web log data has opened new avenues to assist the web administrators and designers to establish adaptive web presence and evolution to fit user requirements.

It is of great importance to process the raw web log data in an appropriate way, and identify the target information intelligently. Huang, An, and Liu focus on exploiting web log sessions, defined as a group of requests made by a single user for a single navigation purpose, in web usage mining. They also compare some of the state-of-the-art techniques in identifying log sessions from Web servers, and present some applications with various types of Web log data.

Yang has observed that it is hard to organize a website such that pages are located where users expect to find them. Through web usages mining, the authors can automatically discover pages in a website whose location

is different from where users expect to find them. This problem of matching website organization with user expectations is pervasive across most websites.

The *Semantic Web technologies* provide several solutions concerning the retrieval of *Semantic Web Documents* (SWDs, mainly, ontologies), which however presuppose that the query is given in a structured way - using a formal language - and provide no advanced means for the (semantic) alignment of the query to the contents of the SWDs. Kotis reports on recent research towards supporting users to form semantic queries – requiring no knowledge and skills for expressing queries in a formal language - and to retrieve SWDs whose content is similar to the queries formed.

Zhu, Nie, and Zhang noticed that extracting object information from the Web is of significant importance. However, the diversity and lack of grammars of Web data make this task very challenging. *Statistical Web object extraction* is a framework based on statistical machine learning theory. The potential advantages of statistical Web object extraction models lie in the fact that Web data have plenty of structure information and the attributes about an object have statistically significant dependencies. These dependencies can be effectively incorporated by developing an appropriate graphical model and thus result in highly accurate extractors.

Borges and Levene advocate the use of *Variable Length Markov Chains* (VLMC) models for Web usage mining since they provide a compact and powerful platform for Web usage analysis. The authors review recent research methods that build VLMC models, as well as methods devised to evaluate both the prediction power and the summarization ability of a VLMC model induced from a collection of navigation sessions. Borges and Levene suggest that due to the well established concepts from Markov chain theory that underpin VLMC models, they will be capable of providing support to cope with the new challenges in Web mining.

With the rapid growth of online information (i.e. web sites, textual document) *text categorization* has become one of the key techniques for handling and organizing data in textual format. First fundamental step in every activity of text analysis is to transform the original file in a classical database, keeping the single words as variables. Cerchiello presents the current state of the art, taking into account all the available classification methods and offering some hints on the more recent approaches. Also, Song discusses issues and methods in automatic *text categorization*, which is the automatic assigning of pre-existing category labels to a group of documents. The article reviews the major models in the field, such as *naïve Bayesian classifiers*, *decision rule classifiers*, the *k-nearest neighbor algorithm*, and *support vector machines*. It also outlines the steps requires to prepare a text classifier and touches on related issues such as dimensionality reduction and machine learning techniques.

Sentiment analysis refers to the classification of texts based on the sentiments they contain. It is an emerging research area in text mining and computational linguistics, and has attracted considerable research attention in the past few years. Leung and Chan introduce a typical sentiment analysis model consisting of three core steps, namely data preparation, review analysis and sentiment classification, and describes representative techniques involved in those steps.

Yu, Tungare, Fan, Pérez-Quiñones, Fox, Cameron, and Cassel describe text classification on a specific information genre, one of the text mining technologies, which is useful in genre-specific information search. Their particular interest is on course syllabus genre. They hope their work is helpful for other genre-specific classification tasks.

Hierarchical models have been shown to be effective in content classification. However, an empirical study has shown that the performance of a hierarchical model varies with given taxonomies; even a semantically sound taxonomy has potential to change its structure for better classification. Tang and Liu elucidate why a given semantics-based hierarchy may not work well in content classification, and how it could be improved for accurate hierarchical classification.

Serrano and Castillo present a survey on the most recent methods to index documents written in natural language to be dealt by text mining algorithms. Although these new indexing methods, mainly based of hyperspaces of word semantic relationships, are a clear improvement on the traditional “bag of words” text representation, they are still producing representations far away from the human mind structures. Future text indexing methods should take more aspects from human mind procedures to gain a higher level of abstraction and semantic depth to success in free-text mining tasks.

Pan presents recent advances in applying machine learning and DM approaches to extract automatically explicit and implicit temporal information from natural language text. The extracted temporal information includes, for example, events, temporal expressions, temporal relations, (vague) event durations, event anchoring, and event orderings.

Saxena, Kothari, and Pandey present a brief survey of various techniques that have been used in the area of *Dimensionality Reduction* (DR). In it, evolutionary computing approach in general, and *Genetic Algorithm* in particular have been used as approach to achieve DR.

Huang, Krneta, Lin, and Wu describe the notion of *Association Bundle Identification*. Association bundles were presented by Huang et al. (2006) as a new pattern of association for DM. On applications such as the *Market Basket Analysis*, association bundles can be compared to, but essentially distinguished from the well-established association rules. Association bundles present meaningful and important associations that association rules unable to identify.

Bartík and Zendulka analyze the problem of association rule mining in relational tables. Discretization of quantitative attributes is a crucial step of this process. Existing discretization methods are summarized. Then, a method called *Average Distance Based Method*, which was developed by the authors, is described in detail. The basic idea of the new method is to separate processing of categorical and quantitative attributes. A new measure called average distance is used during the discretization process.

Leung provides a comprehensive overview of *constraint-based association rule mining*, which aims to find interesting relationships—represented by association rules that satisfy user-specified constraints—among items in a database of transactions. The author describes *what* types of constraints can be specified by users and discusses *how* the properties of these constraints can be exploited for efficient mining of interesting rules.

Pattern discovery was established for second order event associations in early 90's by the authors' research group (Wong, Wang, and Li). A higher order pattern discovery algorithm was devised in the mid 90s for discrete-valued data sets. The discovered high order patterns can then be used for classification. The methodology was later extended to continuous and mixed-mode data. Pattern discovery has been applied in numerous real-world and commercial applications and is an ideal tool to uncover subtle and useful patterns in a database.

Li and Ng discuss the *Positive Unlabelled* learning problem. In practice, it is costly to obtain the class labels for large sets of training examples, and oftentimes the negative examples are lacking. Such practical considerations motivate the development of a new set of classification algorithms that can learn from a set of labeled positive examples P augmented with a set of unlabeled examples U. Four different techniques, S-EM, PEBL, Roc-SVM and LPLP, have been presented. Particularly, LPLP method was designed to address a real-world classification application where the size of positive examples is small.

The classification methodology proposed by Yen aims at using different similarity information matrices extracted from citation, author, and term frequency analysis for scientific literature. These similarity matrices were fused into one generalized similarity matrix by using parameters obtained from a genetic search. The final similarity matrix was passed to an agglomerative hierarchical clustering routine to classify the articles. The work, synergistically integrates multiple similarity information, showed that the proposed method was able to identify the main research disciplines, emerging fields, major contributing authors and their area of expertise within the scientific literature collection.

As computationally intensive experiments are increasingly found to incorporate massive data from multiple sources, the handling of original data, the derived data and all intermediate datasets became challenging. Data provenance is a special kind of Metadata that holds information about who did what and when. Sorathia and Maitra discuss various methods, protocols and system architecture for data provenance. It provides insights about how data provenance can affect decisions for utilization. From recent research perspective, it introduces how grid based data provenance can provide effective solution for data provenance even in *Service Orientation Paradigm*.

The practical usages of *Frequent Pattern Mining* (FPM) algorithms in knowledge mining tasks are still limited due to the lack of interpretability caused from the enormous output size. Conversely, we observed recently a growth of interest in FPM community to summarize the output of an FPM algorithm and obtain a smaller set

of patterns that is non-redundant, discriminative, and representative (of the entire pattern set). Hasan surveys different summarization techniques with a comparative discussion among their benefits and limitations.

Data streams are usually generated in an online fashion characterized by huge volume, rapid unpredictable rates, and fast changing data characteristics. Dang, Ng, Ong, and Lee discuss this challenge in the context of finding frequent sets from transactional data streams. In it, some effective methods are reviewed and discussed, in three fundamental mining models for data stream environments: landmark window, forgetful window and sliding window models.

Research in association rules mining has initially concentrated in solving the obvious problem of finding positive association rules; that is, rules among items that remain in the transactions. It was only several years after that the possibility of finding also negative association rules was investigated, based though on the absence of items from transactions. Ioannis gives an overview of the works having engaged with the subject until now and present a novel view for the definition of negative influence among items, where the choice of one item can trigger the removal of another one.

Lin and Tseng consider mining generalized association rules in an evolving environment. They survey different strategies incorporating the state-of-the-art techniques in dealing with this problem and investigate how to update efficiently the discovered association rules when there is transaction update to the database along with item taxonomy evolution and refinement of support constraint.

Feature extraction/selection has received considerable attention in various areas for which thousands of features are available. The main objective of feature extraction/selection is to identify a subset of feature that are most predictive or informative of a given response variable. Successful implementation of feature extraction/selection not only provides important information for prediction or classification, but also reduces computational and analytical efforts for the analysis of high-dimensional data. Kim presents various feature extraction/selection methods, along with some real examples.

Feature interaction presents a challenge to feature selection for classification. A feature by itself may have little correlation with the target concept, but when it is combined with some other features; they can be strongly correlated with the target concept. Unintentional removal of these features can result in poor classification performance. Handling feature interaction could be computationally intractable. Zhao and Liu provide a comprehensive study for the concept of feature interaction and present several existing feature selection algorithms that apply feature interaction.

François addresses the problem of feature selection in the context of modeling the relationship between explanatory variables and target values, which must be predicted. It introduces some tools, general methodology to be applied on it and identifies trends and future challenges.

Datasets comprising of many features can lead to serious problems, like low classification accuracy. To address such problems, feature selection is used to select a small subset of the most relevant features. The most widely used feature selection approach is the wrapper, which seeks relevant features by employing a classifier in the selection process. Chrysostomou, Lee, Chen, and Liu present the state of the art of the wrapper feature selection process and provide an up-to-date review of work addressing the limitations of the wrapper and improving its performance.

Lisi considers the task of mining multiple-level association rules extended to the more complex case of having an ontology as prior knowledge. This novel problem formulation requires algorithms able to deal actually with ontologies, i.e. without disregarding their nature of logical theories equipped with a formal semantics. Lisi describes an approach that resorts to the methodological apparatus of that logic-based machine learning form known under the name of *Inductive Logic Programming*, and to the expressive power of those knowledge representation frameworks that combine logical formalisms for databases and ontologies.

Arslan presents a unifying view for many sequence alignment algorithms in the literature proposed to guide the alignment process. Guiding finds its true meaning in constrained sequence alignment problems, where constraints require inclusion of known sequence motifs. Arslan summarizes how constraints have evolved from inclusion of simple subsequence motifs to inclusion of subsequences within a tolerance, then to more general regular expression-described motif inclusion, and to inclusion of motifs described by context free grammars.

Xiong, Wang, and Zhang introduce a novel technique to alignment manifolds so as to learn the correspondence relationship in data. The authors argue that it will be more advantageous if they can guide the alignment by relative comparison, which is well defined frequently and easy to obtain. The authors show how this problem can be formulated as an optimization procedure. To make the solution tractable, they further re-formulated it as a *convex semi-definite programming* problem.

Time series data are typically generated by measuring and monitoring applications and plays a central role in predicting the future behavior of systems. Since time series data in its raw form contain no usable structure, it is often segmented to generate a high-level data representation that can be used for prediction. Chundi, and Rosenkrantz discuss the segmentation problem and outline the current state-of-the-art in generating segmentations for the given time series data.

Customer segmentation is the process of dividing customers into distinct subsets (segments or clusters) that behave in the same way or have similar needs. There may exist natural behavioral patterns in different groups of customers or customer transactions. Yang discusses research on using behavioral patterns to segment customers.

Along the lines of Wright and Stashuk, quantization based schemes seemingly discard important data by grouping individual values into relatively large aggregate groups; the use of fuzzy and rough set tools helps to recover a significant portion of the data lost by performing such a grouping. If quantization is to be used as the underlying method of projecting continuous data into a form usable by a discrete-valued knowledge discovery system, it is always useful to evaluate the benefits provided by including a representation of the vagueness derived from the process of constructing the quantization bins.

Lin provides a comprehensive coverage for one of the important problems in DM: sequential pattern mining, especially in the aspect of time constraints. It gives an introduction to the problem, defines the constraints, reviews the important algorithms for the research issue and discusses future trends.

Chen explores the subject of clustering time series, concentrating specially on the area of subsequence time series clustering; dealing with the surprising recent result that the traditional method used in this area is meaningless. He reviews the results that led to this startling conclusion, reviews subsequent work in the literature dealing with this topic, and goes on to argue that two of these works together form a solution to the dilemma.

Qiu and Malthouse summarize the recent developments in cluster analysis for categorical data. The traditional latent class analysis assumes that manifest variables are independent conditional on the cluster identity. This assumption is often violated in practice. Recent developments in latent class analysis relax this assumption by allowing for flexible correlation structure for manifest variables within each cluster. Applications to real datasets provide easily interpretable results.

Learning with Partial Supervision (LPS) aims at combining labeled and unlabeled data to boost the accuracy of classification and clustering systems. The relevance of LPS is highly appealing in applications where only a small ratio of labeled data and a large number of unlabeled data are available. LPS strives to take advantage of traditional clustering and classification machineries to deal with labeled data scarcity. Bouchachia introduces LPS and outlines the different assumptions and existing methodologies concerning it.

Wei, Li, and Li introduce a novel learning paradigm called enclosing machine learning for DM. The new learning paradigm is motivated by two cognition principles of human being, which are cognizing things of the same kind and, recognizing and accepting things of a new kind easily. The authors made a remarkable contribution setting up a bridge that connects the cognition process understanding, with mathematical machine learning tools under the function equivalence framework.

Bouguettaya and Yu focus on investigating the behavior of agglomerative hierarchical algorithms. They further divide these algorithms into two major categories: group based and single-object based clustering methods. The authors choose UPGMA and SLINK as the representatives of each category and the comparison of these two representative techniques could also reflect some similarity and difference between these two sets of clustering methods. Experiment results show a surprisingly high level of similarity between the two clustering techniques under most combinations of parameter settings.

In an effort to achieve improved classifier accuracy, extensive research has been conducted in classifier ensembles. Cluster ensembles offer a solution to challenges inherent to clustering arising from its ill-posed nature. Domeniconi and Razgan discuss recent developments in ensemble methods for clustering.

Tsoumakas and Vlahavas introduce the research area of *Distributed DM* (DDM). They present the state-of-the-art DDM methods for classification, regression, association rule mining and clustering and discuss the application of DDM methods in modern distributed computing environments such as the Grid, peer-to-peer networks and sensor networks.

Wu, Xiong, and Chen highlight the relationship between the clustering algorithms and the distribution of the "true" cluster sizes of the data. They demonstrate that k-means tends to show the uniform effect on clusters, whereas UPGMA tends to take the dispersion effect. This study is crucial for the appropriate choice of the clustering schemes in DM practices.

Huang describes k-modes, a popular DM algorithm for clustering categorical data, which is an extension to k-means with modifications on the distance function, representation of cluster centers and the method to update the cluster centers in the iterative clustering process. Similar to k-means, the k-modes algorithm is easy to use and efficient in clustering large data sets. Other variants are also introduced, including the fuzzy k-modes for fuzzy cluster analysis of categorical data, k-prototypes for clustering mixed data with both numeric and categorical values, and W-k-means for automatically weighting attributes in k-means clustering.

Xiong, Steinbach, Tan, Kumar, and Zhou describe a pattern preserving clustering method, which produces interpretable and usable clusters. Indeed, while there are strong patterns in the data---patterns that may be a key for the analysis and description of the data---these patterns are often split among different clusters by current clustering approaches, since clustering algorithms have no built in knowledge of these patterns and may often have goals that are in conflict with preserving patterns. To that end, their focus is to characterize (1) the benefits of pattern preserving clustering and (2) the most effective way of performing pattern preserving clustering.

Semi-supervised clustering uses the limited background knowledge to aid unsupervised clustering algorithms. Recently, a kernel method for semi-supervised clustering has been introduced. However, the setting of the kernel's parameter is left to manual tuning, and the chosen value can largely affect the quality of the results. Yan and Domeniconi derive a new optimization criterion to automatically determine the optimal parameter of an RBF kernel, directly from the data and the given constraints. The proposed approach integrates the constraints into the clustering objective function, and optimizes the parameter of a *Gaussian* kernel iteratively during the clustering process.

Vilalta and Stepinski propose a new approach to external cluster validation based on modeling each cluster and class as a probabilistic distribution. The degree of separation between both distributions can then be measured using an information-theoretic approach (e.g., relative entropy or *Kullback-Leibler* distance). By looking at each cluster individually, one can assess the degree of novelty (large separation to other classes) of each cluster, or instead the degree of validation (close resemblance to other classes) provided by the same cluster.

Casado, Pacheco, and Nuñez have designed a new technique based on the metaheuristic strategy *Tabu Search* for variable selection for classification, in particular for discriminant analysis and logistic regression. There are very few key references on the selection of variables for their use in discriminant analysis and logistic regression. For this specific purpose only the Stepwise, Backward and Forward methods, can be found in the literature. These methods are simple and they are not very efficient when there are many original variables.

Ensemble learning is an important method of deploying more than one learning model to give improved predictive accuracy for a given learning problem. Rooney, Patterson, and Nugent describe how regression based ensembles are able to reduce the bias and/or variance of the generalization error and review the main techniques that have been developed for the generation and integration of regression based ensembles.

Dominik, Walczak, and Wojciechowski evaluate performance of the most popular and effective classifiers with graph structures, on two kinds of classification problems from different fields of science: computational chemistry, chemical informatics (chemical compounds classification) and information science (web documents classification).

Tong, Koren, and Faloutsos study asymmetric proximity measures on directed graphs, which quantify the relationships between two nodes. Their proximity measure is based on the concept of escape probability. This way, the authors strive to summarize the multiple facets of nodes-proximity, while avoiding some of the pitfalls to which alternative proximity measures are susceptible. A unique feature of the measures is accounting for the underlying directional information. The authors put a special emphasis on computational efficiency, and develop fast solutions that are applicable in several settings and they show the usefulness of their proposed direction-aware proximity method for several applications.

Classification models and in particular binary classification models are ubiquitous in many branches of science and business. Model performance assessment is traditionally accomplishing by using metrics, derived from the confusion matrix or contingency table. It has been observed recently that *Receiver Operating Characteristic* (ROC) curves visually convey the same information as the confusion matrix in much more intuitive and robust fashion. Hamel illustrates how ROC curves can be deployed for model assessment to provide a much deeper and perhaps more intuitive analysis of classification models.

Molecular classification involves the classification of samples into groups of biological phenotypes based on data obtained from microarray experiments. The high-dimensional and multiclass nature of the classification problem demands work on two specific areas: (1) feature selection (FS) and (2) decomposition paradigms. Ooi introduces a concept called *differential prioritization*, which ensures that the optimal balance between two FS criteria, relevance and redundancy, is achieved based on the number of classes in the classification problem.

Incremental learning is a learning strategy that aims at equipping learning systems with adaptively, which allows them to adjust themselves to new environmental conditions. Usually, it implicitly conveys an indication to future evolution and eventually self correction over time as new events happen, new input becomes available, or new operational conditions occur. Bouchachia brings in incremental learning, discusses the main trends of this subject and outlines some of the contributions of the author.

Sheng and Ling introduce the theory of the cost-sensitive learning. The theory focuses on the most common cost (i.e. misclassification cost), which plays the essential role in cost-sensitive learning. Without loss of generality, the authors assume binary classification in this article. Based on the binary classification, they infer that the original cost matrix in real-world applications can always be converted to a simpler one with only false positive and false negative costs.

Thomopoulos focuses on the cooperation of heterogeneous knowledge for the construction of a domain expertise. A two-stage method is proposed: First, verifying expert knowledge (expressed in the conceptual graph model) by experimental data (in the relational model) and second, discovering unexpected knowledge to refine the expertise. A case study has been carried out to further explain the use of this method.

Recupero discusses the graph matching problem and related filtering techniques. It introduces GrepVS, a new fast graph matching algorithm, which combines filtering ideas from other well-known methods in literature. The chapter presents details on hash tables and the Berkeley DB, used to store efficiently nodes, edges and labels. Also, it compares GrepVS filtering and matching phases with the state of the art graph matching algorithms.

Recent technological advances in 3D digitizing, non-invasive scanning, and interactive authoring have resulted in an explosive growth of 3D models. There is a critical need to develop new mining techniques for facilitating the indexing, retrieval, clustering, comparison, and analysis of large collections of 3D models. Shen and Makedon describe a computational framework for mining 3D objects using shape features, and addresses important shape modeling and pattern discovery issues including spherical harmonic surface representation, shape registration, and surface-based statistical inferences. The mining results localize shape changes between groups of 3D objects.

In Zhao and Yao's opinion, while many DM models concentrate on automation and efficiency, interactive DM models focus on adaptive and effective communications between human users and computer systems. The crucial point is not how intelligent users are, or how efficient systems are, but how well these two parts can be connected, adapted, understood and trusted. Some fundamental issues including processes and forms of interactive DM, as well as complexity of interactive DM systems are discussed in this article.

Rivero, Rabuñal, Dorado, and Pazos describe an application of Evolutionary Computation (EC) tools to develop automatically Artificial Neural Networks (ANNs). It also describes how EC techniques have already

been used for this purpose. The technique described in this article allows both design and training of ANNs, applied to the solution of three well-known problems. Moreover, this tool makes the simplification of ANNs to obtain networks with a small number of neurons. Results show how this technique can produce good results in solving DM problems.

Almost all existing DM algorithms have been manually designed. As a result, in general they incorporate human biases and preconceptions in their designs. Freitas and Pappa propose an alternative approach to the design of DM algorithms, namely the automatic creation of DM algorithms by *Genetic Programming* – a type of Evolutionary Algorithm. This approach opens new avenues for research, providing the means to design novel DM algorithms that are less limited by human biases and preconceptions, as well as the opportunity to create automatically DM algorithms tailored to the data being mined.

Gama and Rodrigues present the new model of data gathering from continuous flows of data. What distinguishes current data sources from earlier ones are the continuous flow of data and the automatic data feeds. The authors do not just have people who are entering information into a computer. Instead, they have computers entering data into one another. Major differences are pointed out between this model and previous ones. Also, the incremental setting of learning from a continuous flow of data is introduced by the authors.

The personal name problem is the situation where the authenticity, ordering, gender, and other information cannot be determined correctly and automatically for every incoming personal name. On this paper topics as the evaluation of, and selection from five very different approaches and the empirical comparisons of multiple phonetics and string similarity techniques for the personal name problem, are remarkably addressed by Phua, Lee, and Smith-Miles.

Lo and Khoo present software specification mining, where novel and existing DM and machine learning techniques are utilized to help recover software specifications which are often poorly documented, incomplete, outdated or even missing. These mined specifications can aid software developers in understanding existing systems, reducing software costs, detecting faults and improving program dependability.

Cooper and Zito investigate the statistical properties of the databases generated by the IBM QUEST program. Motivated by the claim (also supported empirical evidence) that item occurrences in real life market basket databases follow a rather different pattern, we propose an alternative model for generating artificial data.

Software metrics-based quality estimation models include those that provide a quality-based classification of program modules and those that provide a quantitative prediction of a quality factor for the program modules. In this article, two count models, *Poisson regression model* (PRM) and *zero-inflated Poisson* (ZIP) regression model, are developed and evaluated by Gao and Khoshgoftaar from those two aspects for a full-scale industrial software system.

Software based on the *Variable Precision Rough Sets* model (VPRS) and incorporating resampling techniques is presented by Griffiths and Beynon as a modern DM tool. The software allows for data analysis, resulting in a classifier based on a set of 'if ... then ...' decision rules. It provides analysts with clear illustrative graphs depicting 'veins' of information within their dataset, and resampling analysis allows for the identification of the most important descriptive attributes within their data.

Program comprehension is a critical task in the software life cycle. Ioannis addresses an emerging field, namely program comprehension through DM. Many researchers consider the specific task to be one of the "hottest" ones nowadays, with large financial and research interest.

The bioinformatics example already approached in the 1e of the present volume is here addressed by Liberati in a novel way, joining two methodologies developed in different fields, namely *minimum description length principle* and *adaptive Bayesian networks*, to implement a new mining tool. The novel approach is then compared with the previous one, showing pros and cons of the two, thus inducing that a combination of the new technique together with the one proposed in the previous edition is the best approach to face the many aspects of the problem.

Integrative analysis of biological data from multiple heterogeneous sources has been employed for a short while with some success. Different DM techniques for such integrative analyses have been developed (which should not be confused with attempts at data integration). Moturu, Parsons, Zhao, and Liu summarize effectively these techniques in an intuitive framework while discussing the background and future trends for this area.

Bhatnagar and Gupta cover in chronological order, the evolution of the formal "KDD process Model", both at the conceptual and practical level. They analyze the strengths and weaknesses of each model and provide the definitions of some of the related terms.

Cheng and Shih present an improved feature reduction method in the combinational input and feature space for *Support Vector Machines* (SVM). In the input space, they select a subset of input features by ranking their contributions to the decision function. In the feature space, features are ranked according to the weighted support vector in each dimension. By combining both input and feature space, Cheng and Shih develop a fast non-linear SVM without a significant loss in performance.

Im and Ras discuss data security in DM. In particular, they describe the problem of confidential data reconstruction by Chase in distributed knowledge discovery systems, and discuss protection methods.

In problems which possibly involve much feature interactions, attribute evaluation measures that estimate the quality of one feature independently of the context of other features measures are not appropriate. Robnik-Šikonja provides and overviews those measures, which are based on the *Relief algorithm*, taking context into account through distance between the instances.

Kretowski and Grzes present an evolutionary approach to induction of decision trees. The evolutionary inducer generates univariate, oblique and mixed trees, and in contrast to classical top-down methods, the algorithm searches for an optimal tree in a global manner. Development of specialized genetic operators allow the system exchange tree parts, generate new sub-trees, prune existing ones as well as change the node type and the tests. A flexible fitness function enables a user to control the inductive biases, and globally induced decision trees are generally simpler with at least the same accuracy as typical top-down classifiers.

Li, Ye, and Kambhamettu present a very general strategy---without assumption of image alignment---for image representation via interest pixel mining. Under the assumption of image alignment, they have intensive studies on *linear discriminant analysis*. One of their papers, "A two-stage linear discriminant analysis via QR-decomposition", was awarded as a fast-breaking paper by Thomson Scientific in April 2007.

As a part of preprocessing and exploratory data analysis, visualization of the data helps to decide which kind of DM method probably leads to good results or whether outliers need to be treated. Rehm, Klawonn, and Kruse present two efficient methods of visualizing high-dimensional data on a plane using a new approach.

Yuan and Wu discuss the problem of repetitive pattern mining in multimedia data. Initially, they explain the purpose of mining repetitive patterns and give examples of repetitive patterns appearing in image/video/audio data accordingly. Finally, they discuss the challenges of mining such patterns in multimedia data, and the differences from mining traditional transaction and text data. The major components of repetitive pattern discovery are discussed, together with the state-of-the-art techniques.

Tsinaraki and Christodoulakis discuss semantic multimedia retrieval and filtering. Since the MPEG-7 is the dominant standard in multimedia content description, they focus on MPEG-7 based retrieval and filtering. Finally, the authors present the *MPEG-7 Query Language* (MP7QL), a powerful query language that they have developed for expressing queries on MPEG-7 descriptions, as well as an MP7QL compatible *Filtering and Search Preferences* (FASP) model. The data model of the MP7QL is the MPEG-7 and its output format is MPEG-7, thus guaranteeing the closure of the language. The MP7QL allows for querying every aspect of an MPEG-7 multimedia content description.

Richard presents some aspects of audio signals automatic indexing with a focus on music signals. The goal of this field is to develop techniques that permit to extract automatically high-level information from the digital raw audio to provide new means to navigate and search in large audio databases. Following a brief overview of audio indexing background, the major building blocks of a typical audio indexing system are described and illustrated with a number of studies conducted by the authors and his colleagues.

With the progress in computing, multimedia data becomes increasingly important to DW. Audio and speech processing is the key to the efficient management and mining of these data. Tan provides in-depth coverage of audio and speech DM and reviews recent advances.

Li presents how DW techniques can be used for improving the quality of association mining. It introduces two important approaches. The first approach requests users to inputs meta-rules through data cubes to describe

desired associations between data items in certain data dimensions. The second approach requests users to provide condition and decision attributes to find desired associations between data granules. The author has made significant contributions to the second approach recently. He is an Associate Editor of the *International Journal of Pattern Recognition and Artificial Intelligence* and an Associate Editor of the *IEEE Intelligent Informatics Bulletin*.

Data cube compression arises from the problem of gaining access and querying massive multidimensional datasets stored in networked data warehouses. Cuzzocrea focuses on state-of-the-art data cube compression techniques and provides a theoretical review of such proposals, by putting in evidence and criticizing the complexities of the building, storing, maintenance, and query phases.

Conceptual modeling is widely recognized to be the necessary foundation for building a database that is well-documented and fully satisfies the user requirements. Although UML and Entity/Relationship are widespread conceptual models, they do not provide specific support for multidimensional modeling. In order to let the user verify the usefulness of a conceptual modeling step in DW design, Golfarelli discusses the expressivity of the *Dimensional Fact Model*, a graphical conceptual model specifically devised for multidimensional design.

Tu introduces the novel technique of automatically tuning database systems based on feedback control loops via rigorous system modeling and controller design. He has also worked on performance analysis of peer-to-peer systems, QoS-aware query processing, and data placement in multimedia databases.

Currently researches focus on particular aspects of a DW development and none of them proposed a systematic design approach that takes into account the end-user requirements. Nabli, Feki, Ben-Abdallah, and Gargouri present a four-step DM/DW conceptual schema design approach that assists the decision maker in expressing their requirements in an intuitive format; automatically transforms the requirements into DM star schemes; automatically merges the star schemes to construct the DW schema; and maps the DW schema to the data source.

Current data warehouses include a time dimension that allows one to keep track of the evolution of measures under analysis. Nevertheless, this dimension cannot be used for indicating changes to dimension data. Malinowski and Zimányi present a conceptual model for designing temporal data warehouses based on the research in temporal databases. The model supports different temporality types, i.e., lifespan, valid time, transaction time coming from source systems, and loading time, generated in a data warehouse. This support is used for representing time-varying levels, dimensions, hierarchies, and measures.

Verykios investigates a representative cluster of research issues falling under the broader area of privacy preserving DM, which refers to the process of mining the data without impinging on the privacy of the data at hand. The specific problem targeted in here is known as association rule hiding and concerns to the process of applying certain types of modifications to the data in such a way that a certain type of knowledge (the association rules) escapes the mining.

The development of DM has the capacity of compromise privacy in ways not previously possible, an issue not only exacerbated through inaccurate data and ethical abuse but also by a lagging legal framework which struggles, at times, to catch up with technological innovation. Wahlstrom, Roddick, Sarre, Estivill-Castro and Vries explore the legal and technical issues of privacy preservation in DM.

Given large data collections of person-specific information, providers can mine data to learn patterns, models, and trends that can be used to provide personalized services. The potential benefits of DM are substantial, but the analysis of sensitive personal data creates concerns about privacy. Oliveira addresses the concerns about privacy, data security, and intellectual property rights on the collection and analysis of sensitive personal data.

With the advent of the information explosion, it becomes crucial to support intelligent personalized retrieval mechanisms for users to identify the results of a manageable size satisfying user-specific needs. To achieve this goal, it is important to model user preference and mine preferences from implicit user behaviors (e.g., user clicks). Hwang discusses recent efforts to extend mining research to preference and identify goals for the future works.

According to González Císaro & Nigro, due to the complexity of nowadays data and the fact that information stored in current databases is not always present at necessary different levels of detail for decision-making processes, a new data type is needed. It is a *Symbolic Object*, which allows representing physics entities or real

word concepts in dual form, respecting their internal variations and structure. The *Symbolic Object Warehouse* permits the intentional description of most important organization concepts, follow by *Symbolic Methods* that work on these objects to acquire new knowledge.

Castillo, Iglesias, and Serrano present a survey on the most known systems to avoid overloading users' mail inbox with unsolicited and illegitimate e-mails. These filtering systems are mainly relying on the analysis of the origin and links contained in e-mails. Since this information is always changing, the systems effectiveness depends on the continuous updating of verification lists.

The evolution of clearinghouses in many ways reflects the evolution of geospatal technologies themselves. The Internet, which has pushed GIS and related technology to the leading edge, has been in many ways fed by the dramatic increase in available data, tools, and applications hosted or developed through the geospatial data clearinghouse movement. Kelly, Haupt, and Baxter outline those advances and offers the reader historic insight into the future of geospatial information.

Angiulli provides an up-to-date view on distance- and density-based methods for large datasets, on subspace outlier mining approaches, and on outlier detection algorithms for processing data streams. Throughout his document different outlier mining tasks are presented, peculiarities of the various methods are pointed out, and relationships among them are addressed. In another paper, Kaur offers various non-parametric approaches used for outlier detection.

The issue of missing values in DM is discussed by Beynon, including the possible drawbacks from their presence, especially when using traditional DM techniques. The nascent CaRBS technique is exposited since it can undertake DM without the need to manage any missing values present. The benchmarked results, when DM incomplete data and data where missing values have been imputed, offers the reader the clearest demonstration of the effect on results from transforming data due to the presence of missing values.

Dorn and Hou examine the quality of association rules derived based on the well-known support-confidence framework using the Chi-squared test. The experimental results show that around 30% of the rules satisfying the minimum support and minimum confidence are in fact statistically insignificant. Integrate statistical analysis into DM techniques can make knowledge discovery more reliable.

The popular querying and data storage models still work with data that are precise. Even though there has recently been much interest in looking at problems arising in storing and retrieving data that are incompletely specified (hence imprecise), such systems have not gained widespread acceptance yet. Nambiar describes challenges involved in supporting imprecision in database systems, briefly explains solutions developed.

Among the different risks Bonafede's work concentrates on operational risks, which form a banking perspective, is due to processes, people, systems (Endogenous) and external events (Exogenous). Bonafede furnishes a conceptual modeling for measurement operational risk and, statistical models applied in the banking sector but adaptable to other fields.

Friedland describes a hidden social structure that may be detectable within large datasets consisting of individuals and their employments or other affiliations. For the most part, individuals in such datasets appear to behave independently. However, sometimes there is enough information to rule out independence and to highlight coordinated behavior. Such individuals acting together are socially tied, and in one case study aimed at predicting fraud in the securities industry, the coordinated behavior was an indicator of higher-risk individuals.

Akdag and Truck focus on studies in *Qualitative Reasoning*, using degrees on a totally ordered scale in a many-valued logic system. Qualitative degrees are a good way to represent uncertain and imprecise knowledge to model approximate reasoning. The qualitative theory takes place between the probability theory and the possibility theory. After defining formalism by logical and arithmetical operators, they detail several aggregators using possibility theory tools such that our probability-like axiomatic system derives interesting results.

Figini presents a comparison, based on survival analysis modeling, between classical and novel DM techniques to predict rates of customer churn. He shows that the novel DM techniques lead to more robust conclusions. In particular, although the lift of the best models are substantially similar, survival analysis modeling gives more valuable information, such as a whole predicted survival function, rather than a single predicted survival probability.

Recent studies show that the method of modeling score distribution is beneficial to various applications. Doloc-Mihu presents the score distribution modeling approach and briefly surveys theoretical and empirical studies on the distribution models, followed by several of its applications.

Valle discusses, among other topics, the most important statistical techniques built to show the relationship between firm performance and its causes, and illustrates the most recent developments in this field.

Data streams arise in many industrial and scientific applications such as network monitoring and meteorology. Dasu and Weiss discuss the unique analytical challenges posed by data streams such as rate of accumulation, continuously changing distributions, and limited access to data. It describes the important classes of problems in mining data streams including data reduction and summarization; change detection; and anomaly and outlier detection. It also provides a brief overview of existing techniques that draw from numerous disciplines such as database research and statistics.

Vast amounts of data are being generated to extract implicit patterns of ambient air pollutant data. Because air pollution data are generally collected in a wide area of interest over a relatively long period, such analyses should take into account both temporal and spatial characteristics. DM techniques can help investigate the behavior of ambient air pollutants and allow us to extract implicit and potentially useful knowledge from complex air quality data. Kim, Temiyasathit, Park, and Chen present the DM processes to analyze complex behavior of ambient air pollution.

Moon, Simpson, and Kumara introduce a methodology for identifying a platform along with variant and unique modules in a product family using design knowledge extracted with an ontology and DM techniques. *Fuzzy c-means clustering* is used to determine initial clusters based on the similarity among functional features. The clustering result is identified as the platform and the modules by the *fuzzy set theory* and classification. The proposed methodology could provide designers with module-based platform and modules that can be adapted to product design during conceptual design.

Analysis of past performance of production systems is necessary in any manufacturing plan to improve manufacturing quality or throughput. However, data accumulated in manufacturing plants have unique characteristics, such as unbalanced distribution of the target attribute, and a small training set relative to the number of input features. Rokach surveys recent researches and applications in this field.

Seng and Srinivasan discuss the numerous challenges that complicate the mining of data generated by chemical processes, which are characterized for being dynamic systems equipped with hundreds or thousands of sensors that generate readings at regular intervals. The two key areas where DM techniques can facilitate knowledge extraction from plant data, namely (1) process visualization and state-identification, and (2) modeling of chemical processes for process control and supervision, are also reviewed in this article.

The telecommunications industry, because of the availability of large amounts of high quality data, is a heavy user of DM technology. Weiss discusses the DM challenges that face this industry and survey three common types of DM applications: marketing, fraud detection, and network fault isolation and prediction.

Understanding the roles of genes and their interactions is a central challenge in genome research. Ye, Janardan, and Kumar describe an efficient computational approach for automatic retrieval of images with overlapping expression patterns from a large database of expression pattern images for *Drosophila melanogaster*. The approach approximates a set of data matrices, representing expression pattern images, by a collection of matrices of low rank through the iterative, approximate solution of a suitable optimization problem. Experiments show that this approach extracts biologically meaningful features and is competitive with other techniques.

Khoury, Toussaint, Ciampi, Antoniano, Murie, and Nadon present, in the context of clustering applied to DNA microarray probes, a better alternative to classical techniques. It is based on proximity-graphs, which has the advantage of being relatively simple and of providing a clear visualization of the data, from which one can directly determine whether or not the data support the existence of clusters.

There has been no formal research about using a *fuzzy Bayesian* model to develop an autonomous task analysis tool. Lin and Lehto summarize a 4-year study that focuses on a Bayesian based machine learning application to help identify and predict the agents' subtasks from the call center's naturalistic decision making's environment. Preliminary results indicate this approach successfully learned how to predict subtasks from the telephone con-

versations and support the conclusion that Bayesian methods can serve as a practical methodology in research area of task analysis as well as other areas of naturalistic decision making.

Financial time series are a sequence of financial data obtained in a fixed period of time. Bose, Leung, and Lau describe how financial time series data can be analyzed using the knowledge discovery in databases framework that consists of five key steps: goal identification, data preprocessing, data transformation, DM, interpretation and evaluation. The article provides an appraisal of several machine learning based techniques that are used for this purpose and identifies promising new developments in hybrid soft computing models.

Maruster and Faber focus on providing insights about patterns of behavior of a specific user group, namely farmers, during the usage of decision support systems. User's patterns of behavior are analyzed by combining these insights with decision making theories, and previous work concerning the development of farmer groups. It provides a method of automatically analyzing the logs resulted from the usage of the decision support system by process mining. The results of their analysis support the redesigning and personalization of decision support systems in order to address specific farmer's characteristics.

Differential proteomics studies the differences between distinct proteomes like normal versus diseased cells, diseased versus treated cells, and so on. Zhang, Orcun, Ouzzani, and Oh introduce the generic DM steps needed for differential proteomics, which include data transformation, spectrum deconvolution, protein identification, alignment, normalization, statistical significance test, pattern recognition, and molecular correlation.

Protein associated data sources such as sequences, structures and interactions accumulate abundant information for DM researchers. Li, Li, Nanyang, and Zhao glimpse the DM methods for the discovery of the underlying patterns at protein interaction sites, the most dominated regions to mediate protein-protein interactions. The authors proposed the concept of binding motif pairs and emerging patterns in DM field.

The applications of DWM are everywhere: from *Applications in Steel Industry* (Ordieres-Meré, Castejón-Limas, and González-Marcos) to *DM in Protein Identification by Tandem Mass Spectrometry* (Wan); from *Mining Smart Card Data from an Urban Transit Network* (Agard, Morency, and Trépanier) to *Data Warehouse in the Finnish Police* (Juntunen)…The list of DWM applications is endless and the future DWM is promising.

Since the current knowledge explosion pushes DWM, a multidisciplinary subject, to ever-expanding new frontiers, any inclusions, omissions, and even revolutionary concepts are a necessary part of our professional life. In spite of all the efforts of our team, should you find any ambiguities or perceived inaccuracies, please contact me at j.john.wang@gmail.com.

REFERENCES

Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., & Uthurusamy, R. (1996). *Advances in knowledge discovery and data mining.* AAAI/MIT Press.

Giudici, P. (2003). *Applied data mining: Statistical methods for business and industry*. John Wiley.

Han, J., & Kamber, M. (2006). *Data mining: Concepts and techniques*. Morgan Kaufmann Publishers.

Hand, D., Mannila, H., & Smyth, P. (2001). *Principles of data mining*. MIT Press.

Turban, E., Aronson, J. E., Liang, T. P., & Sharda, R. (2006). *Decision support systems and business intelligent systems,* 8th edition. Upper Saddle River, NJ: Pearson Prentice Hall.

Wang, J. ed. (2006). *Encyclopedia of data warehousing and mining* (2 Volumes), First Edition. Hershey, PA: Idea Group Reference.

Acknowledgment

The editor would like to thank all of the authors for their insights and excellent contributions to this book. I also want to thank the anonymous reviewers who assisted me in the peer-reviewing process and provided comprehensive, critical, and constructive reviews. Each Editorial Advisory Board member has made a big contribution in terms of guidance and assistance. I owe my thanks to ChunKai Szu and Shaunte Ames, two Editor Assistants, for lending a hand in the whole tedious process.

The editor wishes to acknowledge the help of all involved in the development process of this book, without whose support the project could not have been satisfactorily completed. Fatiha Ouali and Ana Kozyreva, two Graduate Assistants, are hereby graciously acknowledged for their diligent work. A further special note of thanks goes to the staff at IGI Global, whose contributions have been invaluable throughout the entire process, form inception to final publication. Particular thanks go to Kristin M. Roth, Managing Development Editor and Jan Travers, who continuously prodded via e-mail to keep the project on schedule, and to Mehdi Khosrow-Pour, whose enthusiasm motivated me to accept his invitation to join this project.

My appreciation is also due to Montclair State University for awarding me different Faculty Research and Career Development Funds. I would also like to extend my thanks to my brothers Zhengxian Wang, Shubert Wang (an artist, http://www.portraitartist.com/wang/wang.asp), and sister Jixian Wang, who stood solidly behind me and contributed in their own sweet little ways. Thanks go to all Americans, since it would not have been possible for the four of us to come to the U.S. without the support of our scholarships.

Finally, I want to thank my family: my parents, Houde Wang and Junyan Bai for their encouragement; my wife Hongyu Ouyang for her unfailing support, and my sons Leigh Wang and Leon Wang for being without a dad during this project up to two years.

John Wang, PhD
Professor of Information & Decision Sciences
Dept. Management & Information Systems
School of Business
Montclair State University
Montclair, New Jersey, USA

About the Editor

John Wang is a professor in the Department of Management and Information Systems at Montclair State University, USA. Having received a scholarship award, he came to the USA and completed his PhD in operations research from Temple University. Due to his extraordinary contributions beyond a tenured full professor, Dr. Wang has been honored with a special range adjustment in 2006. He has published over 100 refereed papers and six books. He has also developed several computer software programs based on his research findings.

He is the Editor-in-Chief of *International Journal of Applied Management Science, International Journal of Data Analysis Techniques and Strategies, International Journal of Information Systems and Supply Chain Management, International Journal of Information and Decision Sciences*. He is the EiC for the *Advances in Information Systems and Supply Chain Management Book Series*. He has served as a guest editor and referee for many other highly prestigious journals. He has served as track chair and/or session chairman numerous times on the most prestigious international and national conferences.

Also, he is an Editorial Advisory Board member of the following publications: *Intelligent Information Technologies: Concepts, Methodologies, Tools, and Applications, End-User Computing: Concepts, Methodologies, Tools, and Applications, Global Information Technologies: Concepts, Methodologies, Tools, and Applications, Information Communication Technologies: Concepts, Methodologies, Tools, and Applications, Multimedia Technologies: Concepts, Methodologies, Tools, and Applications, Information Security and Ethics: Concepts, Methodologies, Tools, and Applications, Electronic Commerce: Concepts, Methodologies, Tools, and Applications, Electronic Government: Concepts, Methodologies, Tools, and Applications*, etc.

Furthermore, he is the Editor of *Data Warehousing and Mining: Concepts, Methodologies, Tools, and Applications* (six-volume) - http://www.igi-global.com/reference/details.asp?id=6946, and the Editor of the *Encyclopedia of Data Warehousing and Mining*, 1st and 2nd Edition - http://www.igi-pub.com/reference/details.asp?id=7956. His long-term research goal is on the synergy of operations research, data mining and cybernetics. His personal interests include gymnastics, swimming, Wushu (Chinese martial arts), jogging, table-tennis, poetry writing, etc.

Action Rules Mining

Zbigniew W. Ras
University of North Carolina, Charlotte, USA

Elzbieta Wyrzykowska
University of Information Technology and Management, Warsaw, Poland

Li-Shiang Tsay
North Carolina A&T State University, USA

INTRODUCTION

There are two aspects of interestingness of rules that have been studied in data mining literature, objective and subjective measures (Liu et al., 1997), (Adomavicius & Tuzhilin, 1997), (Silberschatz & Tuzhilin, 1995, 1996). Objective measures are data-driven and domain-independent. Generally, they evaluate the rules based on their quality and similarity between them. Subjective measures, including unexpectedness, novelty and actionability, are user-driven and domain-dependent.

A rule is actionable if user can do an action to his/her advantage based on this rule (Liu et al., 1997). This definition, in spite of its importance, is too vague and it leaves open door to a number of different interpretations of actionability. In order to narrow it down, a new class of rules (called action rules) constructed from certain pairs of association rules, has been proposed in (Ras & Wieczorkowska, 2000). Interventions introduced in (Greco et al., 2006) and the concept of information changes proposed in (Skowron & Synak, 2006) are conceptually very similar to action rules. Action rules have been investigated further in (Wang at al., 2002), (Tsay & Ras, 2005, 2006), (Tzacheva & Ras, 2005), (He at al., 2005), (Ras & Dardzinska, 2006), (Dardzinska & Ras, 2006), (Ras & Wyrzykowska, 2007). To give an example justifying the need of action rules, let us assume that a number of customers have closed their accounts at one of the banks. We construct, possibly the simplest, description of that group of people and next search for a new description, similar to the one we have, with a goal to identify a new group of customers from which no-one left that bank. If these descriptions have a form of rules, then they can be seen as actionable rules. Now, by comparing these two descriptions, we may find the cause why these accounts have been closed and formulate an action which if undertaken by the bank, may prevent other customers from closing their accounts. Such actions are stimulated by action rules and they are seen as precise hints for actionability of rules. For example, an action rule may say that by inviting people from a certain group of customers for a glass of wine by a bank, it is guaranteed that these customers will not close their accounts and they do not move to another bank. Sending invitations by regular mail to all these customers or inviting them personally by giving them a call are examples of an action associated with that action rule.

In (Tzacheva & Ras, 2005) the notion of a cost and feasibility of an action rule was introduced. The cost is a subjective measure and feasibility is an objective measure. Usually, a number of action rules or chains of action rules can be applied to re-classify a certain set of objects. The cost associated with changes of values within one attribute is usually different than the cost associated with changes of values within another attribute. The strategy for replacing the initially extracted action rule by a composition of new action rules, dynamically built and leading to the same reclassification goal, was proposed in (Tzacheva & Ras, 2005). This composition of rules uniquely defines a new action rule. Objects supporting the new action rule also support the initial action rule but the cost of reclassifying them is lower or even much lower for the new rule. In (Ras & Dardzinska, 2006) authors present a new algebraic-type top-down strategy for constructing action rules from single classification rules. Algorithm ARAS, proposed in (Ras & Wyrzykowska, 2007), is a bottom-up strategy generating action rules. In (He at al., 2005) authors give a strategy for discovering action rules directly from a database.

BACKGROUND

In the paper by (Ras & Wieczorkowska, 2000), the notion of an action rule was introduced. The main idea was to generate, from a database, special type of rules which basically form a hint to users showing a way to reclassify objects with respect to some distinguished attribute (called a decision attribute). Clearly, each relational schema gives a list of attributes used to represent objects stored in a database. Values of some of these attributes, for a given object, can be changed and this change can be influenced and controlled by user. However, some of these changes (for instance "profit") can not be done directly to a decision attribute. In such a case, definitions of this decision attribute in terms of other attributes (called classification attributes) have to be learned. These new definitions are used to construct action rules showing what changes in values of some attributes, for a given class of objects, are needed to reclassify objects the way users want. But, users may still be either unable or unwilling to proceed with actions leading to such changes. In all such cases, we may search for definitions of values of any classification attribute listed in an action rule. By replacing a value of such attribute by its definition, we construct new action rules which might be of more interest to business users than the initial rule. Action rules can be constructed from pairs of classification rules, from a single classification rule, and directly from a database.

MAIN THRUST OF THE CHAPTER

The technology dimension will be explored to clarify the meaning of actionable rules including action rules and action rules schema.

Action Rules Discovery in Information Systems

An information system is used for representing knowledge. Its definition, given here, is due to (Pawlak, 1991).

By an information system we mean a pair S = (U, A), where:

1. U is a nonempty, finite set of objects (object identifiers),
2. A is a nonempty, finite set of attributes i.e. $a:U \rightarrow V_a$ for $a \in A$, where V_a is called the domain of a.

Information systems can be seen as decision tables. In any decision table together with the set of attributes a partition of that set into conditions and decisions is given. Additionally, we assume that the set of conditions is partitioned into stable and flexible conditions (Ras & Wieczorkowska, 2000).

Attribute $a \in A$ is called stable for the set U if its values assigned to objects from U can not be changed in time. Otherwise, it is called flexible. "Date of Birth" is an example of a stable attribute. "Interest rate" on any customer account is an example of a flexible attribute. For simplicity reason, we will consider decision tables with only one decision. We adopt the following definition of a decision table:

By a decision table we mean an information system $S = (U, A_{St} \cup A_{Fl} \cup \{d\})$, where $d \notin A_{St} \cup A_{Fl}$ is a distinguished attribute called decision. The elements of A_{St} are called stable conditions, whereas the elements of $A_{Fl} \cup \{d\}$ are called flexible conditions. Our goal is to change values of attributes in A_{Fl} for some objects from U so values of the attribute d for these objects may change as well. A formal expression describing such a property is called an action rule (Ras & Wieczorkowska, 2000), (Tsay & Ras, 2005).

To construct an action rule (Tsay & Ras, 2005), let us assume that two classification rules, each one referring to a different decision class, are considered. We assume here that these two rules have to be equal on their stable attributes, if they are both defined on them. We use Table 1 to clarify the process of action rule construction. Here, "St" means stable attribute and "Fl" means flexible one.

In a standard representation, these two classification rules have a form:

$$r1 = [\, a1 \wedge b1 \wedge c1 \wedge e1 \rightarrow d1 \,], \; r2 = [\, a1 \wedge b2 \wedge g2 \wedge h2 \rightarrow d2 \,].$$

Assume now that object x supports rule r1 which means that x is classified as d1. In order to reclassify x to class d2, we need to change its value b from b1 to b2 but also we have to require that g(x)=g2 and that the value h for object x has to be changed to h2. This is the meaning of the (r1,r2)-action rule r defined by the expression below:

Table 1. Two classification rules extracted from S

a (St)	b (Fl)	c (St)	e (Fl)	g (St)	h (Fl)	d (Decision)
a1	b1	c1	e1			d1
a1	b2			g2	h2	d2

$r = [[a1 \wedge g2 \wedge (b, b1 \rightarrow b2) \wedge (h, \rightarrow h2)] \Rightarrow (d, d1 \rightarrow d2)]$.

The term $[a1 \wedge g2]$ is called the header of the action rule. Assume now that by Sup(t) we mean the number of tuples having property t. By the support of (r1,r2)-action rule r we mean: $Sup[a1 \wedge b1 \wedge g2 \wedge d1]$. Action rule schema associated with rule r2 is defined as:

$[[a1 \wedge g2 \wedge (b, \rightarrow b2) \wedge (h, \rightarrow h2)] \Rightarrow (d, d1 \rightarrow d2)]$.

By the confidence Conf(r) of (r1,r2)-action rule r we mean:

$[Sup[a1 \wedge b1 \wedge g2 \wedge d1]/ Sup[a1 \wedge b1 \wedge g2]] \bullet [Sup[a1 \wedge b2 \wedge c1 \wedge d2]/ Sup[a1 \wedge b2 \wedge c1]]$.

System DEAR (Tsay & Ras, 2005) discovers action rules from pairs of classification rules.

Actions Rules Discovery, a New Simplified Strategy

A bottom-up strategy, called ARAS, generating action rules from single classification rules was proposed in (Ras & Wyrzykowska, 2007). We give an example describing its main steps.

Let us assume that the decision system S = (U, $A_{St} \cup A_{Fl} \cup \{d\}$), where U = {x1,x2,x3,x4,x5,x6,x7,x8}, is represented by Table 2. A number of different methods can be used to extract rules in which the THEN part consists of the decision attribute d and the IF part consists of attributes belonging to $A_{St} \cup A_{Fl}$. In our example, the set A_{St} ={a,b,c} contains stable attributes and A_{Fl} = {e,f,g} contains flexible attributes. System LERS (Grzymala-Busse, 1997) is used to extract classification rules.

We are interested in reclassifying d2-objects either to class d1 or d3. Four certain classification rules describing either d1 or d3 are discovered by LERS from the decision system S. They are given below:

$r1 = [b1 \wedge c1 \wedge f2 \wedge g1] \rightarrow d1$, $r2 = [a2 \wedge b1 \wedge e2 \wedge f2] \rightarrow d3$, $r3 = e1 \rightarrow d1$, $r4 = [b1 \wedge g2] \rightarrow d3$.

Action rule schemas associated with r1, r2, r3, r4 and the reclassification task either $(d, d2 \rightarrow d1)$ or $(d, d2 \rightarrow d3)$ are:

$r1[d2 \rightarrow d1] = [b1 \wedge c1 \wedge (f, \rightarrow f2) \wedge (g, \rightarrow g1)] \Rightarrow (d, d2 \rightarrow d1)$,
$r2[d2 \rightarrow d3] = [a2 \wedge b1 \wedge (e, \rightarrow e2) \wedge (f, \rightarrow f2)] \Rightarrow (d, d2 \rightarrow d3)$,
$r3[d2 \rightarrow d1] = [(e, \rightarrow e1)] \Rightarrow (d, d2 \rightarrow d1)$,
$r4[d2 \rightarrow d3] = [b1 \wedge (g, \rightarrow g2)] \Rightarrow (d, d2 \rightarrow d3)$.

We can show that $Sup(r1[d2 \rightarrow d1])$= {x3, x6, x8}, $Sup(r2[d2 \rightarrow d3])$= {x6, x8}, $Sup(r3[d2 \rightarrow d1])$= {x3,x4,x5,x6,x7,x8}, $Sup(r4[d2 \rightarrow d3])$ = {x3,x4,x6,x8}.

Assuming that $U[r1,d2] = Sup(r1[d2 \rightarrow d1])$, $U[r2,d2] = Sup(r2[d2 \rightarrow d3])$, $U[r3,d2] = Sup(r3[d2 \rightarrow d1])$, $U[r4,d2] = Sup(r4[d2 \rightarrow d3])$ and by applying ARAS algorithm we get:

$[b1 \wedge c1 \wedge a1]^* = \{x1\} \not\subset U[r1,d2]$, $[b1 \wedge c1 \wedge a2]^* = \{x6, x8\} \subseteq U[r1,d2]$,
$[b1 \wedge c1 \wedge f3]^* = \{x6\} \subseteq U[r1,d2]$, $[b1 \wedge c1 \wedge g2]^* = \{x2, x7\} \not\subset U[r1,d2]$,
$[b1 \wedge c1 \wedge g3]^* = \{x3, x8\} \subseteq U[r1,d2]$.

ARAS will construct two action rules for the first action rule schema:

Table 2. Decision system

U	a	b	c	e	f	g	d
x1	a1	b1	c1	e1	f2	g1	d1
x2	a2	b1	c2	e2	f2	g2	d3
x3	a3	b1	c1	e2	f2	g3	d2
x4	a1	b1	c2	e2	f2	g1	d2
x5	a1	b2	c1	e3	f2	g1	d2
x6	a2	b1	c1	e2	f3	g1	d2
x7	a2	b3	c2	e2	f2	g2	d2
x8	a2	b1	c1	e3	f2	g3	d2

$[b1 \wedge c1 \wedge (f, f3 \rightarrow f2) \wedge (g, \rightarrow g1)] \Rightarrow (d, d2 \rightarrow d1)$,
$[b1 \wedge c1 \wedge (f, \rightarrow f2) \wedge (g, g3 \rightarrow g1)] \Rightarrow (d, d2 \rightarrow d1)$.

In a similar way we construct action rules from the remaining three action rule schemas.

ARAS consists of two main modules. To explain them in a better way, we use another example which has no connection with Table 2. The first module of ARAS extracts all classification rules from S following LERS strategy. Assuming that d is the decision attribute and user is interested in reclassifying objects from its value d1 to d2, we treat the rules defining d1 as seeds and build clusters around them. For instance, if A_{St} ={a, b, g} and A_{Fl} = {c, e, h} are attributes in S = $(U, A_{St} \cup A_{Fl} \cup \{d\})$, and r =$[[a1 \wedge b1 \wedge c1 \wedge e1] \rightarrow d1]$ is a classification rule in S, where Va={a1,a2,a3}, Vb={b1,b2,b3}, Vc={c1,c2,c3}, Ve={e1,e2,e3}, Vg={g1,g2,g3}, Vh={h1,h2,h3}, then we remove from S all tuples containing values a2, a3, b2, b3, c1, e1 and we use again LERS to extract rules from the obtained subsystem.

Each rule defining d2 is used jointly with r to construct an action rule. The validation step of each of the set-inclusion relations, in the second module of ARAS, is replaced by checking if the corresponding term was marked by LERS in the first module of ARAS.

FUTURE TRENDS

Business user may be either unable or unwilling to proceed with actions leading to desired reclassifications of objects. Undertaking the actions may be trivial, feasible to an acceptable degree, or may be practically very difficult. Therefore, the notion of a cost of an action rule is of very great importance. New strategies for discovering action rules of the lowest cost either in an autonomous information system or a distributed one, based on ontologies, should be investigated.

(He et al., 2005) proposed a strategy for discovering action rules directly from a database. More research needs to be done also in that area.

CONCLUSION

Attributes are divided into two groups: stable and flexible. By stable attributes we mean attributes which values can not be changed (for instance, age or maiden name). On the other hand attributes (like percentage rate or loan approval to buy a house) which values can be changed are called flexible. Rules are extracted from a decision table, using standard KD methods, with preference given to flexible attributes - so mainly they are listed in a classification part of rules. Most of these rules can be seen as actionable rules and the same used to construct action-rules.

REFERENCES

Adomavicius, G., Tuzhilin, A. (1997). Discovery of actionable patterns in databases: the action hierarchy approach, Proceedings of KDD97 Conference, Newport Beach, CA, AAAI Press.

Dardzinska, A., Ras, Z. (2006). Cooperative discovery of interesting action rules, *Proceedings of FQAS 2006 Conference*, Milano, Italy, (Eds. H.L. Larsen et al.), Springer, LNAI 4027, 489-497.

Greco, S., Matarazzo, B., Pappalardo, N., Slowinski, R. (2005). Measuring expected effects of interventions based on decision rules, *Journal of Experimental and Theoretical Artificial Intelligence 17 (1-2)*, Taylor and Francis.

Grzymala-Busse, J. (1997). A new version of the rule induction system LERS, *Fundamenta Informaticae 31 (1),* 27-39.

He, Z., Xu, X., Deng, S., Ma, R. (2005). Mining action rules from scratch, in *Expert Systems with Applications* 29 (3), Elsevier, 691-699.

Liu, B., Hsu, W., Chen, S. (1997). Using general impressions to analyze discovered classification rules, *Proceedings of KDD97 Conference*, Newport Beach, CA, AAAI Press.

Pawlak, Z., (1991). Rough Sets: Theoretical aspects of reasoning about data, Kluwer.

Ras, Z., Dardzinska, A. (2006). Action Rules Discovery, a new simplified strategy, in *Foundations of Intelligent Systems, Proceedings of ISMIS'06,* F. Esposito et al. (Eds.), Bari, Italy, LNAI 4203, Springer, 445-453.

Ras, Z., Wieczorkowska, A. (2000). Action Rules: how to increase profit of a company, in *Principles of Data*

Mining and Knowledge Discovery, (Eds. D.A. Zighed, J. Komorowski, J. Zytkow), *Proceedings of PKDD'00*, Lyon, France, LNAI 1910, Springer, 587-592.

Ras, Z., Wyrzykowska, E. (2007). ARAS: Action rules discovery based on agglomerative strategy, in *Mining Complex Data, Post-Proceedings of the ECML/PKDD'07 Third International Workshop, MCD 2007*, LNAI, Springer, will appear.

Silberschatz, A., Tuzhilin, A. (1995). On subjective measures of interestingness in knowledge discovery, *Proceedings of KDD'95 Conference*, AAAI Press.

Silberschatz, A., Tuzhilin, A. (1996). What makes patterns interesting in knowledge discovery systems, *IEEE Transactions on Knowledge and Data Engineering 5 (6)*.

Skowron, A., Synak, P. (2006). Planning Based on Reasoning about Information Changes, in *Rough Sets and Current Trends in Computing*, LNCS 4259, Springer, 165-173.

Tsay, L.-S., Ras, Z. (2005). Action Rules Discovery System DEAR, Method and Experiments, *Journal of Experimental and Theoretical Artificial Intelligence 17 (1-2)*, Taylor and Francis, 119-128.

Tsay, L.-S., Ras, Z. (2006). Action Rules Discovery System DEAR3, in *Foundations of Intelligent Systems, Proceedings of ISMIS'06*, F. Esposito et al. (Eds.), Bari, Italy, LNAI 4203, Springer, 483-492.

Tzacheva, A., Ras, Z. (2005). Action rules mining, *International Journal of Intelligent Systems 20 (7),* Wiley, 719-736.

Wang, K., Zhou, S., Han, J. (2002). Profit mining: From patterns to actions, in *Proceedings of EDBT'02*, 70-87.

KEY TERMS

A

Actionable Rule: A rule is actionable if user can do an action to his/her advantage based on this rule.

Autonomous Information System: Information system existing as an independent entity.

Domain of Rule: Attributes listed in the IF part of a rule.

Flexible Attribute: Attribute is called flexible if its value can be changed in time.

Knowledge Base: A collection of rules defined as expressions written in predicate calculus. These rules have a form of associations between conjuncts of values of attributes.

Ontology: An explicit formal specification of how to represent objects, concepts and other entities that are assumed to exist in some area of interest and relationships holding among them. Systems that share the same ontology are able to communicate about domain of discourse without necessarily operating on a globally shared theory. System commits to ontology if its observable actions are consistent with definitions in the ontology.

Stable Attribute: Attribute is called stable for the set U if its values assigned to objects from U can not change in time.

Active Learning with Multiple Views

Ion Muslea
SRI International, USA

INTRODUCTION

Inductive learning algorithms typically use a set of labeled examples to learn class descriptions for a set of user-specified concepts of interest. In practice, labeling the training examples is a tedious, time consuming, error-prone process. Furthermore, in some applications, the labeling of each example also may be extremely expensive (e.g., it may require running costly laboratory tests). In order to reduce the number of labeled examples that are required for learning the concepts of interest, researchers proposed a variety of methods, such as active learning, semi-supervised learning, and meta-learning.

This article presents recent advances in reducing the need for labeled data in multi-view learning tasks; that is, in domains in which there are several disjoint subsets of features (views), each of which is sufficient to learn the target concepts. For instance, as described in Blum and Mitchell (1998), one can classify segments of televised broadcast based either on the video or on the audio information; or one can classify Web pages based on the words that appear either in the pages or in the hyperlinks pointing to them. In summary, this article focuses on using multiple views for active learning and improving multi-view active learners by using semi-supervised- and meta-learning.

BACKGROUND

Active, Semi-Supervised, and Multi-view Learning

Most of the research on multi-view learning focuses on semi-supervised learning techniques (Collins & Singer, 1999, Pierce & Cardie, 2001) (i.e., learning concepts from a few labeled and many unlabeled examples). By themselves, the unlabeled examples do not provide any direct information about the concepts to be learned. However, as shown by Nigam, et al. (2000) and Raskutti, et al. (2002), their distribution can be used to boost the accuracy of a classifier learned from the few labeled examples.

Intuitively, semi-supervised, multi-view algorithms proceed as follows: first, they use the small labeled training set to learn one classifier in each view; then, they bootstrap the views from each other by augmenting the training set with unlabeled examples on which the other views make high-confidence predictions. Such algorithms improve the classifiers learned from labeled data by also exploiting the implicit' information provided by the distribution of the unlabeled examples.

In contrast to semi-supervised learning, active learners (Tong & Koller, 2001) typically detect and ask the user to label only the most informative examples in the domain, thus reducing the user's data-labeling burden. Note that active and semi-supervised learners take different approaches to reducing the need for labeled data; the former explicitly search for a minimal set of labeled examples from which to perfectly learn the target concept, while the latter aim to improve a classifier learned from a (small) set of labeled examples by exploiting some additional unlabeled data.

In keeping with the active learning approach, this article focuses on minimizing the amount of labeled data without sacrificing the accuracy of the learned classifiers. We begin by analyzing co-testing (Muslea, 2002), which is a novel approach to active learning. Co-testing is a multi-view active learner that maximizes the benefits of labeled training data by providing a principled way to detect the most informative examples in a domain, thus allowing the user to label only these.

Then, we discuss two extensions of co-testing that cope with its main limitations—the inability to exploit the unlabeled examples that were not queried and the lack of a criterion for deciding whether a task is appropriate for multi-view learning. To address the former, we present Co-EMT (Muslea et al., 2002a), which interleaves co-testing with a semi-supervised, multi-view learner. This hybrid algorithm combines the benefits of active and semi-supervised learning by

detecting the most informative examples, while also exploiting the remaining unlabeled examples. Second, we discuss Adaptive View Validation (Muslea et al., 2002b), which is a meta-learner that uses the experience acquired while solving past learning tasks to predict whether multi-view learning is appropriate for a new, unseen task.

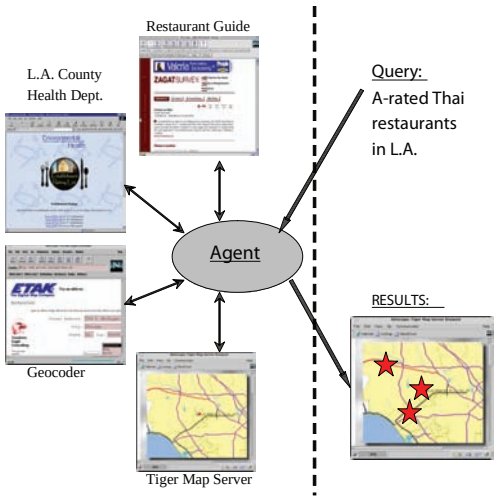
A Motivating Problem: Wrapper Induction

Information agents such as Ariadne (Knoblock et al., 2001) integrate data from pre-specified sets of Web sites so that they can be accessed and combined via database-like queries. For example, consider the agent in Figure 1, which answers queries such as the following:

Show me the locations of all** Thai **restaurants in** L.A. **that are** A-rated **by the L.A. County Health Department.

To answer this query, the agent must combine data from several Web sources:

Figure 1. An information agent that combines data from the Zagat's restaurant guide, the L.A. County Health Department, the ETAK Geocoder, and the Tiger Map service



- From Zagat's, it obtains the name and address of all Thai restaurants in L.A.
- From the L.A. County Web site, it gets the health rating of any restaurant of interest.
- From the Geocoder, it obtains the latitude/longitude of any physical address.
- From Tiger Map, it obtains the plot of any location, given its latitude and longitude.

Information agents typically rely on *wrappers* to extract the useful information from the relevant Web pages. Each wrapper consists of a set of extraction rules and the code required to apply them. As manually writing the extraction rules is a time-consuming task that requires a high level of expertise, researchers designed wrapper induction algorithms that learn the rules from user-provided examples (Muslea et al., 2001).

In practice, information agents use hundreds of extraction rules that have to be updated whenever the format of the Web sites changes. As manually labeling examples for each rule is a tedious, error-prone task, one must learn high accuracy rules from just a few labeled examples. Note that both the small training sets and the high accuracy rules are crucial to the successful deployment of an agent. The former minimizes the amount of work required to create the agent, thus making the task manageable. The latter is required in order to ensure the quality of the agent's answer to each query: when the data from multiple sources is integrated, the errors of the corresponding extraction rules get compounded, thus affecting the quality of the final result; for instance, if only 90% of the Thai restaurants and 90% of their health ratings are extracted correctly, the result contains only 81% (90% x 90% = 81%) of the A-rated Thai restaurants.

We use wrapper induction as the motivating problem for this article because, despite the practical importance of learning accurate wrappers from just a few labeled examples, there has been little work on active learning for this task. Furthermore, as explained in Muslea (2002), existing general-purpose active learners cannot be applied in a straightforward manner to wrapper induction.

MAIN THRUST

In the context of wrapper induction, we intuitively describe three novel algorithms: Co-Testing, Co-EMT,

and Adaptive View Validation. Note that these algorithms are *not* specific to wrapper induction, and they have been applied to a variety of domains, such as text classification, advertisement removal, and discourse tree parsing (Muslea, 2002).

Co-Testing: Multi-View Active Learning

Co-Testing (Muslea, 2002, Muslea et al., 2000), which is the first multi-view approach to active learning, works as follows:

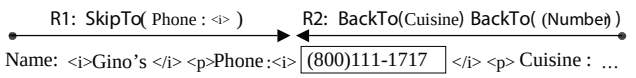
- First, it uses a small set of labeled examples to learn one classifier in each view.
- Then, it applies the learned classifiers to all unlabeled examples and asks the user to label one of the examples on which the views predict different labels.
- It adds the newly labeled example to the training set and repeats the whole process.

Intuitively, Co-Testing relies on the following observation: if the classifiers learned in each view predict a different label for an unlabeled example, at least one of them makes a mistake on that prediction. By asking the user to label such an example, Co-Testing is guaranteed to provide useful information for the view that made the mistake.

To illustrate Co-Testing for wrapper induction, consider the task of extracting restaurant phone numbers from documents similar to the one shown in Figure 2. To extract this information, the wrapper must detect both the beginning and the end of the phone number. For instance, to find where the phone number begins, one can use the following rule:

R1 = *SkipTo*(**Phone:<i>**)

Figure 2. The forward rule R1 and the backward rule R2 detect the beginning of the phone number. Forward and backward rules have the same semantics and differ only in terms of from where they are applied (start/end of the document) and in which direction



This rule is applied *forward*, from the beginning of the page, and it ignores everything until it finds the string **Phone:<i>**. Note that this is not the only way to detect where the phone number begins. An alternative way to perform this task is to use the following rule:

R2 = *BackTo*(**Cuisine**) *BackTo*((***Number***))

which is applied *backward*, from the end of the document. R2 ignores everything until it finds "Cuisine" and then, again, skips to the first number between parentheses.

Note that R1 and R2 represent descriptions of the same concept (i.e., beginning of phone number) that are learned in two different views (see Muslea et al. [2001] for details on learning forward and backward rules). That is, views V1 and V2 consist of the sequences of characters that precede and follow the beginning of the item, respectively. View V1 is called the forward view, while V2 is the backward view. Based on V1 and V2, Co-Testing can be applied in a straightforward manner to wrapper induction. As shown in Muslea (2002), Co-Testing clearly outperforms existing state-of-the-art algorithms, both on wrapper induction and a variety of other real world domains.

Co-EMT: Interleaving Active and Semi-Supervised Learning

To further reduce the need for labeled data, Co-EMT (Muslea et al., 2002a) combines active and semi-supervised learning by interleaving Co-Testing with Co-EM (Nigam & Ghani, 2000). Co-EM, which is a semi-supervised, multi-view learner, can be seen as the following iterative, two-step process: first, it uses the hypotheses learned in each view to probabilistically label all the unlabeled examples; then it learns a new hypothesis in each view by training on the probabilistically labeled examples provided by the other view.

By interleaving active and semi-supervised learning, Co-EMT creates a powerful synergy. On one hand, Co-Testing boosts Co-EM's performance by providing it with highly informative labeled examples (instead of random ones). On the other hand, Co-EM provides Co-Testing with more accurate classifiers (learned from both labeled and unlabeled data), thus allowing Co-Testing to make more informative queries.

Co-EMT was not yet applied to wrapper induction, because the existing algorithms are not probabilistic

learners; however, an algorithm similar to Co-EMT was applied to information extraction from free text (Jones et al., 2003). To illustrate how Co-EMT works, we describe now the generic algorithm Co-EMT^{WI}, which combines Co-Testing with the semi-supervised wrapper induction algorithm described next.

In order to perform semi-supervised wrapper induction, one can exploit a third view, which is used to evaluate the confidence of each extraction. This new content-based view (Muslea et al., 2003) describes the actual item to be extracted. For example, in the phone numbers extraction task, one can use the labeled examples to learn a simple grammar that describes the field content: (*Number*) *Number – Number*. Similarly, when extracting URLs, one can learn that a typical URL starts with the string "http://www.'', ends with the string ".html'', and contains no HTML tags.

Based on the forward, backward, and content-based views, one can implement the following semi-supervised wrapper induction algorithm. First, the small set of labeled examples is used to learn a hypothesis in each view. Then, the forward and backward views feed each other with unlabeled examples on which they make high-confidence extractions (i.e., strings that are extracted by either the forward or the backward rule and are also compliant with the grammar learned in the third, content-based view).

Given the previous Co-Testing and the semi-supervised learner, Co-EMT^{WI} combines them as follows. First, the sets of labeled and unlabeled examples are used for semi-supervised learning. Second, the extraction rules that are learned in the previous step are used for Co-Testing. After making a query, the newly labeled example is added to the training set, and the whole process is repeated for a number of iterations. The empirical study in Muslea, et al., (2002a) shows that, for a large variety of text classification tasks, Co-EMT outperforms both Co-Testing and the three state-of-the-art semi-supervised learners considered in that comparison.

View Validation: Are the Views Adequate for Multi-View Learning?

The problem of *view validation* is defined as follows: given a new unseen multi-view learning task, how does a user choose between solving it with a multi- or a single-view algorithm? In other words, how does one know whether multi-view learning will outperform pooling all features together and applying a single-view learner? Note that this question must be answered while having access to just a few labeled and many unlabeled examples: applying both the single- and multi-view active learners and comparing their relative performances is a self-defeating strategy, because it doubles the amount of required labeled data (one must label the queries made by both algorithms).

The need for view validation is motivated by the following observation: while applying Co-Testing to dozens of extraction tasks, Muslea et al. (2002b) noticed that the forward and backward views are appropriate for most, but not all, of these learning tasks. This view adequacy issue is related tightly to the best extraction accuracy reachable in each view. Consider, for example, an extraction task in which the forward and backward rules lead to a high- and low-accuracy rule, respectively. Note that Co-Testing is not appropriate for solving such tasks; by definition, multi-view learning applies only to tasks in which each view is sufficient for learning the target concept (obviously, the low-accuracy view is insufficient for accurate extraction).

To cope with this problem, one can use Adaptive View Validation (Muslea et al., 2002b), which is a meta-learner that uses the experience acquired while solving past learning tasks to predict whether the views of a new unseen task are adequate for multi-view learning. The view validation algorithm takes as input several solved extraction tasks that are labeled by the user as having views that are adequate or inadequate for multi-view learning. Then, it uses these solved extraction tasks to learn a classifier that, for new unseen tasks, predicts whether the views are adequate for multi-view learning.

The (meta-) features used for view validation are properties of the hypotheses that, for each solved task, are learned in each view (i.e., the percentage of unlabeled examples on which the rules extract the same string, the difference in the complexity of the forward and backward rules, the difference in the errors made on the training set, etc.). For both wrapper induction and text classification, Adaptive View Validation makes accurate predictions based on a modest amount of training data (Muslea et al., 2002b).

FUTURE TRENDS

There are several major areas of future work in the field of multi-view learning. First, there is a need for a

view detection algorithm that automatically partitions a domain's features in views that are adequate for multi-view learning. Such an algorithm would remove the last stumbling block against the wide applicability of multi-view learning (i.e., the requirement that the user provides the views to be used). Second, in order to reduce the computational costs of active learning (re-training after each query is CPU-intensive), one must consider look-ahead' strategies that detect and propose (near) optimal sets of queries. Finally, Adaptive View Validation has the limitation that it must be trained separately for each application domain (e.g., once for wrapper induction, once for text classification, etc.). A major improvement would be a *domain-independent* view validation algorithm that, once trained on a mixture of tasks from various domains, can be applied to any new learning task, independently of its application domain.

CONCLUSION

In this article, we focus on three recent developments that, in the context of multi-view learning, reduce the need for labeled training data.

- **Co-Testing:** A general-purpose, multi-view active learner that outperforms existing approaches on a variety of real-world domains.
- **Co-EMT:** A multi-view learner that obtains a robust behavior over a wide spectrum of learning tasks by interleaving active and semi-supervised multi-view learning.
- **Adaptive View Validation:** A meta-learner that uses past experiences to predict whether multi-view learning is appropriate for a new unseen learning task.

REFERENCES

Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. *Proceedings of the Conference on Computational Learning Theory (COLT-1998)*.

Collins, M., & Singer, Y. (1999). Unsupervised models for named entity classification. *Empirical Methods in Natural Language Processing & Very Large Corpora* (pp. 100-110).

Jones, R., Ghani, R., Mitchell, T., & Riloff, E. (2003). Active learning for information extraction with multiple view feature sets. *Proceedings of the ECML-2003 Workshop on Adaptive Text Extraction and Mining.*

Knoblock, C. et al. (2001). The Ariadne approach to Web-based information integration. *International Journal of Cooperative Information Sources*, *10*, 145-169.

Muslea, I. (2002). *Active learning with multiple views* [doctoral thesis]. Los Angeles: Department of Computer Science, University of Southern California.

Muslea, I., Minton, S., & Knoblock, C. (2000). Selective sampling with redundant views. *Proceedings of the National Conference on Artificial Intelligence (AAAI-2000)*.

Muslea, I., Minton, S., & Knoblock, C. (2001). Hierarchical wrapper induction for semi-structured sources. *Journal of Autonomous Agents & Multi-Agent Systems*, *4*, 93-114.

Muslea, I., Minton, S., & Knoblock, C. (2002a). Active + semi-supervised learning = robust multi-view learning. *Proceedings of the International Conference on Machine Learning (ICML-2002)*.

Muslea, I., Minton, S., & Knoblock, C. (2002b). Adaptive view validation: A first step towards automatic view detection. *Proceedings of the International Conference on Machine Learning (ICML-2002)*.

Muslea, I., Minton, S., & Knoblock, C. (2003). Active learning with strong and weak views: A case study on wrapper induction. *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-2003)*.

Nigam, K., & Ghani, R. (2000). Analyzing the effectiveness and applicability of co-training. *Proceedings of the Conference on Information and Knowledge Management (CIKM-2000)*.

Nigam, K., McCallum, A., Thrun, S., & Mitchell, T. (2000). Text classification from labeled and unlabeled documents using EM. *Machine Learning*, *39*(2-3), 103-134.

Pierce, D., & Cardie, C. (2001). Limitations of co-training for natural language learning from large datasets. *Empirical Methods in Natural Language Processing*, 1-10.

Raskutti, B., Ferra, H., & Kowalczyk, A. (2002). Using unlabeled data for text classification through addition of cluster parameters. *Proceedings of the International Conference on Machine Learning (ICML-2002)*.

Tong, S., & Koller, D. (2001). Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, *2*, 45-66.

KEY TERMS

Active Learning: Detecting and asking the user to label only the most informative examples in the domain (rather than randomly-chosen examples).

Inductive Learning: Acquiring concept descriptions from labeled examples.

Meta-Learning: Learning to predict the most appropriate algorithm for a particular task.

Multi-View Learning: Explicitly exploiting several disjoint sets of features, each of which is sufficient to learn the target concept.

Semi-Supervised Learning: Learning from both labeled and unlabeled data.

View Validation: Deciding whether a set of views is appropriate for multi-view learning.

Wrapper Induction: Learning (highly accurate) rules that extract data from a collection of documents that share a similar underlying structure.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 12-16, copyright 2005 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

Adaptive Web Presence and Evolution through Web Log Analysis

Xueping Li
University of Tennessee, Knoxville, USA

INTRODUCTION

The Internet has become a popular medium to disseminate information and a new platform to conduct electronic business (e-business) and electronic commerce (e-commerce). With the rapid growth of the WWW and the intensified competition among the businesses, effective web presence is critical to attract potential customers and retain current customer thus the success of the business. This poses a significant challenge because the web is inherently dynamic and web data is more sophisticated, diverse, and dynamic than traditional well-structured data. Web mining is one method to gain insights into how to evolve the web presence and to ultimately produce a predictive model such that the evolution of a given web site can be categorized under its particular context for strategic planning. In particular, web logs contain potentially useful information and the analysis of web log data have opened new avenues to assist the web administrators and designers to establish adaptive web presence and evolution to fit user requirements.

BACKGROUND

People have realized that web access logs are a valuable resource for discovering various characteristics of customer behaviors. Various data mining or machine learning techniques are applied to model and understand the web user activities (Borges and Levene, 1999; Cooley et al., 1999; Kosala et al., 2000; Srivastava et al., 2000; Nasraoui and Krishnapuram, 2002). The authors in (Kohavi, 2001; Mobasher et al., 2000) discuss the pros and cons of mining the e-commerce log data. Lee and Shiu (Lee and Shiu, 2004) propose an adaptive website system to automatically change the website architecture according to user browsing activities and to improve website usability from the viewpoint of efficiency. Recommendation systems are used by an ever-increasing number of e-commerce sites to help consumers find products to purchase (Schafer et al, 2001). Specifically, recommendation systems analyze the users' and communities' opinions and transaction history in order to help individuals identify products that are most likely to be relevant to their preferences (e.g. Amazon.com, eBay.com). Besides web mining technology, some researches investigate on Markov chain to model the web user access behavior (Xing et al., 2002; Dhyani et al., 2003; Wu et al., 2005). Web log analysis is used to extract terms to build web page index, which is further combined with text-based and anchor-based indices to improve the performance of the web site search (Ding and Zhou, 2007). A genetic algorithm is introduced in a model-driven decision-support system for web site optimization (Asllani and Lari, 2007). A web forensic framework as an alternative structure for clickstream data analysis is introduced for customer segmentation development and loyal customer identification; and some trends in web data analysis are discussed (Sen et al., 2006).

MAIN FOCUS

Broadly speaking, web log analysis falls into the range of web usage mining, one of the three categories of web mining (Kosala and Blockeel, 2000; Srivastava et al., 2002). There are several steps involved in web log analysis: web log acquisition, cleansing and preprocessing, and pattern discovery and analysis.

Web Log Data Acquisition

Web logs contain potentially useful information for the study of the effectiveness of web presence. Most websites enable logs to be created to collect the server and client activities such as access log, agent log, error log, and referrer log. Access logs contain the bulk of data including the date and time, users' IP addresses,

requested URL, and so on. Agent logs provide the information of the users' browser type, browser version, and operating system. Error logs provide problematic and erroneous links on the server such as "file not found", "forbidden to access", et al. Referrer logs provide information about web pages that contain the links to documents on the server.

Because of the stateless characteristic of the Hyper Text Transfer Protocol (HTTP), the underlying protocol used by the WWW, each request in the web log seems independent of each other. The identification of user sessions, in which all pages that a user requests during a single visit, becomes very difficulty (Cooley et al., 1999). Pitkow (1995, 1997, 1998) pointed out that local caching and proxy servers are two main obstacles to get reliable web usage data. Most browsers will cache the recently pages to improve the response time. When a user clicks the "back" button in a browser, the cached document is displayed instead of retrieving the page from the web server. This process can not be recorded by the web log. The existence of proxy servers makes it even harder to identify the user session. In the web server log, requests from a proxy server will have the same identifier although the requests may come from several different users. Because of the cache ability of proxy servers, one requested page in web server logs may actually be viewed by several users. Besides the above two obstacles, the dynamic content pages such as Active Server Pages (ASP) and Java Server Pages (JSP) will also create problems for web logging. For example, although the same Uniform Resource Locator (URL) appears in a web server log, the content that is requested by users might be totally different.

To overcome the above obstacles of inaccuracy web log resulting from caching, proxy server and dynamic web pages, specialized logging techniques are needed. One way is to configure the web server to customize the web logging. Another is to integrate the web logging function into the design of the web pages. For example, it is beneficial to an e-commerce web site to log the customer shopping cart information which can be implemented using ASP or JSP. This specialized log can record the details that the users add items to or remove items from their shopping carts thus to gain insights into the user behavior patterns with regard to shopping carts.

Besides web server logging, package sniffers and cookies can be used to further collection web log data. Packet sniffers can collect more detailed information than web server log by looking into the data packets transferred on the wire or air (wireless connections). However, it suffers from several drawbacks. First, packet sniffers can not read the information of encrypted data. Second, it is expensive because each server needs a separate packet sniffer. It would be difficult to manage all the sniffers if the servers are located in different geographic locations. Finally, because the packets need to be processed by the sniffers first, the usage of packet sniffers may reduce the performance of the web servers. For these reasons, packet sniffing is not widely used as web log analysis and other data collecting techniques.

A cookie is a small piece of information generated by the web server and stored at the client side. The client first sends a request to a web server. After the web server processes the request, the web server will send back a response containing the requested page. The cookie information is sent with the response at the same time. The cookie typically contains the session id, expiration date, user name and password and so on. This information will be stored at the client machine. The cookie information will be sent to the web server every time the client sends a request. By assigning each visitor a unique session id, it becomes easy to identify the sessions. However, some users prefer to disable the usage of cookies on their computers which limits the wide application of cookies.

Web Log Data Cleansing and Preprocessing

Web log data cleansing and preprocessing is critical to the success of the web log analysis. Even though most of the web logs are collected electronically, serious data quality issues may arise from a variety of sources such as system configuration, software bugs, implementation, data collection process, and so on. For example, one common mistake is that the web logs collected from different sites use different time zone. One may use Greenwich Mean Time (GMT) while the other uses Eastern Standard Time (EST). It is necessary to cleanse the data before analysis.

There are some significant challenges related to web log data cleansing. One of them is to differentiate the web traffic data generated by web bots from that generated by "real" web visitors. Web bots, including web robots and

spiders/crawlers, are automated programs that browse websites. Examples of web bots include Google Crawler (Brin and Page, 1998), Ubicrawler (Boldi et al. 2004), and Keynote (www.keynote.com). The traffic from the web bots may tamper the visiting statistics, especially in the e-commerce domain. Madsen (Madsen, 2002) proposes a page tagging method of clickstream collection through the execution of JavaScript at the client's browsers. Other challenges include the identification of sessions and unique customers.

Importing the web log into traditional database is another way to preprocess the web log and to allow further structural queries. For example, web access log data can be exported to a database. Each line in the access log represents a single request for a document on the web server. The typical form of an access log of a request is as follows:

hostname - - [dd/Mon/yyyy:hh24:mm:ss tz] request status bytes

An example is:

uplherc.upl.com - - [01/Aug/1995:00:00:07 -0400] "GET / HTTP/1.0" 304 0

which is from the classical data collected from the web server at the NASA's Kennedy Space Center. Each entry of the access log consists of several fields. The meaning of each field is as following:

- **Host name**: A hostname when possible; otherwise, the IP address if the host name could not be looked up.
- **Timestamp**: In the format " *dd/Mon/yyyy:hh24:mm:ss tz* ", where *dd* is the day of the month, *Mon* is the abbreviation name of the month, *yyyy* is the year, *hh24:mm:ss* is the time of day using a 24-hour clock, and *tz* stands for time zone as shown in the example "*[01/Aug/1995:00:00:07 -0400]*". For consistency, hereinafter we use "day/month/year" date format.
- **Request:** Requests are given in quotes, for example "*GET / HTTP/1.0*". Inside the quotes, "*GET*" is the HTTP service name, "*/*" is the request object, and "*HTTP/1.0*" is the HTTP protocol version.
- **HTTP reply code:** The status code replied by the web server. For example, a reply code "200" means the request is successfully processed. The detailed description about HTTP reply codes refers to RFC (http://www.ietf.org/rfc).
- **Reply Bytes:** This field shows the number of bytes replied.

In the above example, the request came from the host "*uplherc.upl.com*" at *01/Aug/1995:00:00:07*. The requested document was the root homepage "/". The status code was "*304*" which meant that the client copy of document was up to date and thus "*0*" bytes were responded to the client. Then, each entry in the access log can be mapped into a field of a table in a database for query and pattern discovery.

Pattern Discovery and Analysis

A variety of methods and algorithms have been developed in the fields of statistics, pattern recognition, machine learning and data mining (Fayyad et al., 1994; Duda et al., 2000). This section describes the techniques that can be applied in the web log analysis domain.

(1) **Statistical Analysis –** It is the most common and simple yet effective method to explore the web log data and extract knowledge of user access patterns which can be used to improve the design of the web site. Different descriptive statistical analyses, such as mean, standard deviation, median, frequency, and so on, can be performed on variables including number of requests from hosts, size of the documents, server reply code, requested size from a domain, and so forth.

There are a few interesting discoveries about web log data through statistical analysis. Recently, the power law distribution has been shown to apply to the web traffic data in which the probability $P(x)$ that a performance measure x decays as a power law, following $P(x) \sim x^{-\alpha}$. A few power law distributions have been discovered: the number of visits to a site (Adamic et al., 1999), the number of page within a site (Huberman et al., 1999), and the number of links to a page (Albert et al., 1999; Barabási et al., 1999).

Given the highly uneven distribution of the documents request, the e-commerce websites should adjust the caching policy to improve the visitor's experience. C. Cunha (1997) point out that small images account for the majority of the

traffic. It would be beneficial if the website can cache these small size documents in memory. For e-commerce websites, the highly populated items should be arranged to allow fast access because these items will compose over 50% of the total requests. These insights are helpful for the better design and adaptive evolution of the web sites.

2. **Clustering and Classification –** Techniques to group a set of items with similar characteristics and/or to map them into predefined classes. In the web log analysis domain, there are two major clusters of interest to discover: web usage clustering and web pages clustering. Clustering of web usage can establish the groups of users that exhibit similar browsing behaviors and infer user demographic information. Such knowledge is especially useful for marketing campaign in e-commerce applications and personalized web presence. On the other hand, clustering analysis of the web pages can discover the web pages with related content. This is useful for the development of Internet search engine. Classification can be accomplished through well developed data mining algorithms including Bayesian classifier, k-nearest neighbor classifier, support vector machines, and so on (Duda et al., 2000).
3. **Associative Rules** – Associative rules mining is to find interesting associations or correlations among large data sets. In the web log mining domain, one is interested in discovering the implications or correlations of user access patterns. For example, users who access page *A* also visit page *B*; customers who purchase product *C* also purchase product *D*. A typical associative rule application is market basket analysis. This knowledge is useful for effective web presence and evolution by laying out user friendly hyper links for easier access. It can help for e-commerce web site to promote products as well.
4. **Sequential Patterns** – The sequential patterns mining attempts to find inter-transaction patterns such that the presence of one event is followed by another (Mannila et al., 1995, Srikant and Agrawal, 1996). In the context of web log analysis, the discovery of sequential patterns helps to predict user visit patterns and to target certain groups based on these patterns.

FUTURE TRENDS

With the explosive growth of the Internet and ever increasing popularity of e-commerce, privacy is becoming a sensitive topic that attracts many research efforts. How to make sure the identity of an individual is not compromised while effective web log analysis can be conducted is a big challenge. An initiative called Platform for Privacy Preference (P3P) is ongoing at the World Wide Web Consortium (W3C). How to analyze the web log online and make timely decision to update and evolve the web sites is another promising topic.

CONCLUSION

An effective web presence is crucial to enhance the image of a company, increase the brand and product awareness, provide customer services, and gather information. The better understanding of the web's topology and user access patterns, along with modeling and designing efforts, can help to develop search engines and strategies to evolve the web sites. Web logs contain potentially useful information for the study of the effectiveness of web presence. The components of web log analysis are described in this chapter. The approaches and challenges of acquisition and preprocessing of web logs are presented. Pattern discovery techniques including statistical analysis, clustering and classification, associative rules and sequential pattern are discussed in the context of web log analysis towards adaptive web presence and evolution.

REFERENCES

Adamic, L.A. and Huberman, B.A. (1999). The Nature of Markets in the World Wide Web. *Computing in Economics and Finance*, no. 521.

Albert, R., Jeong, H. and Barabási, A.L. (1999). The Diameter of the World Wide Web. *Nature*, 401:130-130.

Asllani A. and Lari A. (2007). Using genetic algorithm for dynamic and multiple criteria web-site optimizations. *European Journal of Operational Research*, 176(3): 1767-1777.

Barabási, A. and Albert, R. (1999). Emergence of Scaling in Random Networks. *Science*, 286(5439): 509 – 512.

Boldi, P., Codenotti, B., Santini, M., and Vigna, S. (2004a). UbiCrawler: a scalable fully distributed Web crawler. *Software, Practice and Experience*, 34(8):711–726.

Borges, J. and Levene, M. (1999). Data mining of user navigation patterns, in: H.A. Abbass, R.A. Sarker, C. Newton (Eds.). Web Usage Analysis and User Profiling, *Lecture Notes in Computer Science*, Springer-Verlag, pp: 92–111.

Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117.

Cooley, R., Mobashar, B. and Shrivastava, J. (1999). Data Preparation for Mining World Wide Web Browsing Patterns. *Knowledge and Information System*, 1(1): 5-32.

Cunha, C. (1997). Trace Analysis and Its Application to Performance Enhancements of Distributed Information Systems. Doctoral thesis, Department of Computer Science, Boston University.

Ding, C. and Zhou, J. (2007). Information access and retrieval: Log-based indexing to improve web site search. *Proceedings of the 2007 ACM symposium on Applied computing SAC '07*, 829-833.

Dhyani, D., Bhowmick, S. and Ng, Wee-Kong (2003). Modeling and Predicting Web Page Accesses Using Markov Processes. *14th International Workshop on Database and Expert Systems Applications* (DEXA'03), p.332.

Duda, R. O., Hart P. E., and Stork, D. G. (2000). Pattern Classification. John Wiley & Sons, Inc.

Fayyad, U., Piatetsky-Shaprio G., and Smyth P. (1994) From data mining to knowledge discovery: an overview. *In Proc. ACM KDD*.

Huberman, B.A. and Adamic, L.A. (1999). Growth Dynamics of the World Wide Web. *Nature*, 401: 131.

Kohavi, R. (2001). Mining E-Commerce Data: The Good, the Bad, and the Ugly. *KDD' 2001* Industrial Track, San Francisco, CA.

Kosala, R. and Blockeel H. (2000). Web Mining Research: A Survey. SIGKDD: *SIGKDD Explorations: Newsletter of the Special Interest Group (SIG) on Knowledge Discovery & Data Mining*, 2(1): 1- 15.

Lee, J.H. and Shiu, W.K. (2004). An adaptive website system to improve efficiency with web mining techniques. *Advanced Engineering Informatics*, 18: 129-142.

Madsen, M.R. (2002). Integrating Web-based Clickstream Data into the Data Warehouse. *DM Review Magazine*, August, 2002.

Mannila H., Toivonen H., and Verkamo A. I. (1995). Discovering frequent episodes in sequences. In *Proc. of the First Int'l Conference on Knowledge Discovery and Data Mining*, pp. 210-215, Montreal, Quebec.

Nasraoui, O. and Krishnapuram R. (2002). One step evolutionary mining of context sensitive associations and web navigation patterns. *SIAM Conference on Data Mining*, Arlington, VA, pp: 531–547.

Pitkow, J.E. (1995). Characterizing browsing strategies in the World Wide Web. *Computer Networks and ISDN Systems*, 27(6): 1065-1073.

Pitkow, J.E. (1997). In search of reliable usage data on the WWW. *Computer Networks and ISDN Systems*, 29(8): 1343-1355.

Pitkow, J.E. (1998). Summary of WWW characterizations. *Computer Networks and ISDN Systems*, 30(1-7): 551-558.

Schafer, J.B., Konstan A.K. and Riedl J. (2001). E-Commerce Recommendation Applications. *Data Mining and Knowledge Discovery*, 5(1-2): 115-153.

Sen, A., Dacin P. A. and Pattichis C. (2006). Current trends in web data analysis. *Communications of the ACM*, 49(11): 85-91.

Srikant R. and Agrawal R. (1996). Mining sequential patterns: Generalizations and performance improvements. In *Proc. of the Fifth Int'l Conference on Extending Database Technology*, Avignon, France.

Srivastava, J., Cooley, R., Deshpande, M. and Tan, P.-N. 2000. Web usage mining: discovery and applications of usage patterns from web data. *SIGKDD Explorations*, 1(2):12-23.

Srivastava, J., Desikan P., and Kumar V. (2002). Web Mining: Accomplishments and Future Directions. *Proc. US Nat'l Science Foundation Workshop on Next-Generation Data Mining* (NGDM).

Wu, F., Chiu, I. and Lin., J. 2005. Prediction of the Intention of Purchase of the user Surfing on the Web Using Hidden Markov Model. *Proceedings of ICSSSM*, 1: 387-390.

Xing, D. and Shen, J. 2002. A New Markov Model For Web Access Prediction. *Computing in Science and Engineering*, 4(6): 34 – 39.

KEY TERMS

Web Access Log: Access logs contain the bulk of data including the date and time, users' IP addresses, requested URL, and so on. The format of the web log varies depending on the configuration of the web server.

Web Agent Log: Agent logs provide the information of the users' browser type, browser version, and operating system.

Web Error Log: Error logs provide problematic and erroneous links on the server such as "file not found", "forbidden to access", et al. and can be used to diagnose the errors that the web serve encounters in processing the requests.

Web Log Acquisition: The process of obtaining the web log information. The web logs can be recorded through the configuration of the web server.

Web Log Analysis: The process of parsing the log files from a web server to derive information about the user access patterns and how the server processes the requests. It helps to assist the web administrators to establish effective web presence, assess marketing promotional campaigns, and attract customers.

Web Log Pattern Discovery: The process of application of data mining techniques to discover the interesting patterns from the web log data.

Web Log Preprocessing and Cleansing: The process of detecting and removing inaccurate web log records that arise from a variety of sources such as system configuration, software bugs, implementation, data collection process, and so on.

Web Presence: A collection of web files focusing on a particular subject that is presented on a web server on the World Wide Web.

Web Referrer Log: Referrer logs provide information about web pages that contain the links to documents on the server.

Web Usage Mining: The subfield of web mining that aims at analyzing and discovering interesting patterns of web server log data.

Aligning the Warehouse and the Web

Hadrian Peter
University of the West Indies, Barbados

Charles Greenidge
University of the West Indies, Barbados

INTRODUCTION

Data warehouses have established themselves as necessary components of an effective IT strategy for large businesses. To augment the streams of data being siphoned from transactional/operational databases warehouses must also integrate increasing amounts of external data to assist in decision support. Modern warehouses can be expected to handle up to 100 Terabytes or more of data. (Berson and Smith, 1997; Devlin, 1998; Inmon 2002; Imhoff et al, 2003; Schwartz, 2003; Day 2004; Peter and Greenidge, 2005; Winter and Burns 2006; Ladley, 2007).

The arrival of newer generations of tools and database vendor support has smoothed the way for current warehouses to meet the needs of the challenging global business environment (Kimball and Ross, 2002; Imhoff et al, 2003; Ross, 2006).

We cannot ignore the role of the Internet in modern business and the impact on data warehouse strategies. The web represents the richest source of external data known to man (Zhenyu et al, 2002; Chakrabarti, 2002; Laender et al, 2002) but we must be able to couple raw text or poorly structured data on the web with descriptions, annotations and other forms of summary meta-data (Crescenzi et al, 2001).

In recent years the Semantic Web initiative has focussed on the production of "smarter data". The basic idea is that instead of making programs with near human intelligence, we rather carefully add meta-data to existing stores so that the data becomes "marked up" with all the information necessary to allow not-so-intelligent software to perform analysis with minimal human intervention. (Kalfoglou et al, 2004)

The Semantic Web builds on established building block technologies such as Unicode, URIs(Uniform Resource Indicators) and XML (Extensible Markup Language) (Dumbill, 2000; Daconta et al, 2003; Decker et al, 2000). The modern data warehouse must embrace these emerging web initiatives. In this paper we propose a model which provides mechanisms for sourcing external data resources for analysts in the warehouse.

BACKGROUND

Data Warehousing

Data warehousing is an evolving IT strategy in which data is periodically siphoned off from multiple heterogeneous operational databases and composed in a specialized database environment for business analysts posing queries. Traditional data warehouses tended to focus on historical/archival data but modern warehouses are required to be more nimble, utilizing data which becomes available within days of creation in the operational environments (Schwartz , 2003; Imhoff et al, 2003; Strand and Wangler, 2004; Ladley, 2007). Data warehouses must provide different views of the data, allowing users the options to "drill down" to highly granular data or to produce highly summarized data for business reporting. This flexibility is supported by the use of robust tools in the warehouse environment (Berson and Smith, 1997; Kimball and Ross, 2002).

Data Warehousing accomplishes the following:

- Facilitates ad hoc end-user querying
- Facilitates the collection and merging of large volumes of data
- Seeks to reconcile the inconsistencies and fix the errors that may be discovered among data records
- Utilizes meta-data in an intensive way.
- Relies on an implicit acceptance that external data is readily available

Some major issues in data warehousing design are:

- Ability to handle vast quantities of data
- Ability to view data at differing levels of granularity
- Query Performance versus ease of query construction by business analysts
- Ensuring Purity, Consistency and Integrity of data entering warehouse
- Impact of changes in the business IT environments supplying the warehouse
- Costs and Return-on-Investment (ROI)

External Data and Search Engines

External data is an often ignored but essential ingredient in the decision support analysis performed in the data warehouse environment. Relevant sources such as trade journals, news reports and stock quotes are required by warehouse decision support personnel when reaching valid conclusions based on internal data (Inmon, 2002; Imhoff et al, 2003).

External data, if added to the warehouse, may be used to put into context data originating from operational systems. The web has long provided a rich source of external data, but robust Search Engine (SE) technologies must be used to retrieve this data (Chakrabarti, 2002; Sullivan, 2000). In our model we envisage a cooperative nexus between the data warehouse and search engines. We introduce a special intermediate and independent data staging layer called the meta-data engine (M-DE).

Search Engines are widely recognized as imperfect yet practical tools to access global data via the Internet. Search Engines continue to mature with new regions, such as the Deep Web, once inaccessible, now becoming accessible (Bergman, 2001; Wang and Lochovsky, 2003; Zillman, 2005). The potential of current and future generations of SEs for harvesting huge tracts of external data cannot be underestimated.

Our model allows a naïve (business) user to pose a query which can be modified to target the domain(s) of interest associated with the user. The SE acts on the modified query to produce results. Once results are retrieved from the SE there is a further processing stage to format the results data for the requirements of the data warehouse.

MAIN THRUST

A

Detailed Model

We now examine the contribution of our model. In particular we highlight the Query Modifying Filter (QMF), Search Engines submission and retrieval phases, and meta-data engine components. The approaches taken in our model aims to enhance the user experience while maximizing the efficiency in the search process.

A query modification process is desirable due to the intractable nature of composing queries. We also wish to target several different search engines with our queries. We note that search engines may independently provide special operators and/or programming tools (e.g. Google API) to allow for tweaking of the default operations of the engine. Thus the Query Modifying Filter (labeled filter in the diagram) may be used to fine tune a generic query to meet the unique search features of a particular search engine. We may need to enhance terms supplied by a user to better target the domain(s) of a user. Feedback from the meta-data engine can be used to guide the development of the Query Modifying Filter.

The use of popular search engines in our suite guarantees the widest possible coverage by our engine. The basic steps in the querying process is:

1. Get user's (naïve) query
2. Apply QMF to produce several modified, search engine specific queries
3. Submit modified queries to their respective search engines
4. Retrieve results and form seed links
5. Use seed links and perform depth/breadth first traversals using seed links
6. Store results from step. 5 to disk

Architecture

For effective functioning, our proposed system must address a number of areas pertaining to both the data warehouse and SE environments, namely:

1. Relevance of retrieved data to a chosen domain
2. Unstructured/semi-structured nature of data on the web
3. Analysis & Generation of meta-data

4. Granularity
5. Temporal Constraints (time stamps, warehouse cycles etc.)
6. Data Purity

Our model bridges the disparate worlds of the warehouse and the web by applying maturing technologies while making key observations about the data warehouse and search engine domains. Directly introducing an integrated search engine into the data warehouse environment, albeit a quick fix solution, would have serious limitations. We would be confronted with a problematic and incongruous situation in which highly structured data in the warehouse would be brought in contact with web data which is often unstructured or semi-structured. We can make far fewer assumptions about unstructured data when compared with structured data.

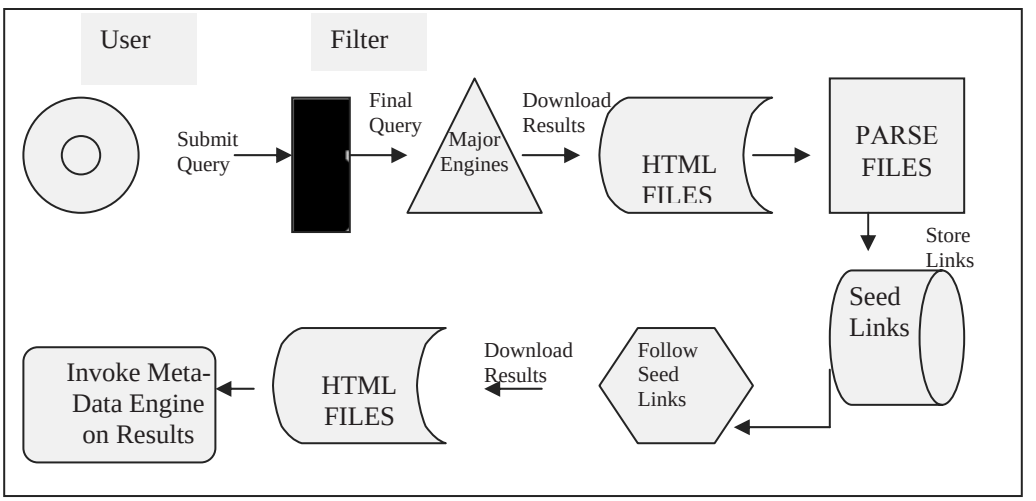
A standard SE ordinarily consists of two parts; a crawler program, and an indexing program. Meta-search engines function by querying other search engines and then ranking combined results in order of relevance. In our model we take the meta-search approach instead of initiating a separate crawler and then utilize the meta-data engine components to assume the role of an indexing program. The meta-data engine forms a bridge between the warehouse and search engine environments.

The M-DE in our model provides an interface in which initially poorly structured or semi-structured data becomes progressively more structured, through the generation of meta-data, to conform to the processing requirements of the data warehouse.

Research is currently very active in the use of XML and related technologies, which can encode web data with the necessary labeling from the time the data is generated. Older, pre-existing web data may also be retrofitted with necessary meta-data. The Semantic Web initiative holds promise that some day most web data will be transparent and readable by intelligent agent software. Closely associated with the Semantic Web is the concept of web services in which business software interoperates using special online directories and protocols. In the worse case we may have to resort to traditional Information Retrieval (IR), Natural Language Processing (NLP), Machine Learning (ML) and other Artificial Intelligence (AI) techniques to grapple with latent semantic issues in the free text (Shah et al, 2002; Laender et al, 2002; Hassell et al, 2006; Holzinger et al, 2006).

The architecture of the M-DE allows for a variety of technologies to be applied. Data on the Web covers a wide continuum including free text in natural language, poorly structured data, semi-structured data, and also highly structured data. Perversely, highly structured data may yet be impenetrable if the structure is unknown, as in the case with some data existing in Deep Web databases (Wang and Lochovsky, 2003; Zillman, 2005).

Figure 1. Query and retrieval in hybrid search engine



We now examine the M-DE component operation in detail. Logically we divide the model into four components. In the diagram these are labeled Filter, Modify, Analyze and Format respectively.

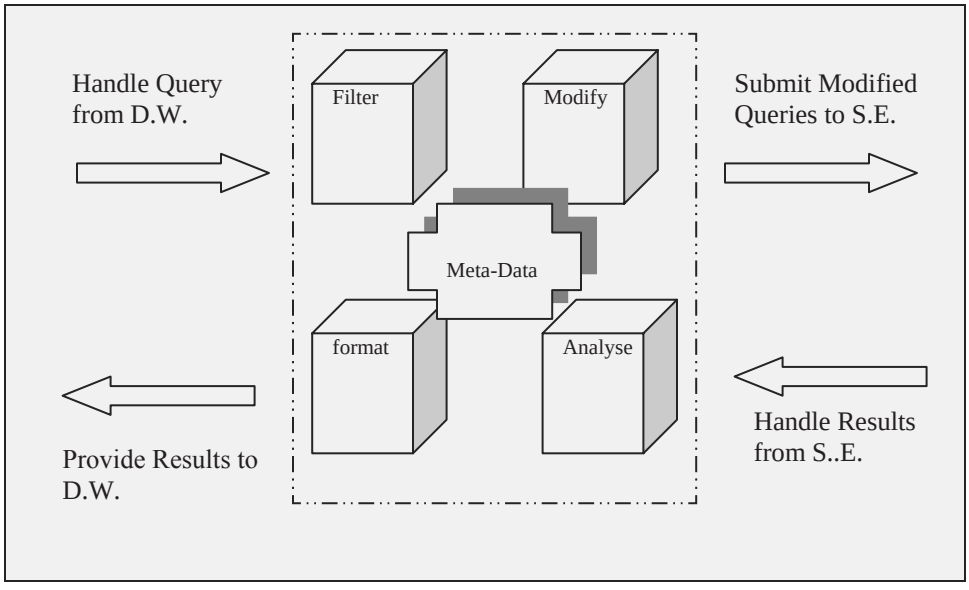
Firstly, the Filter component takes a query from a user and checks that it is valid and suitable for further action. In some cases the user is directed immediately to existing results, or may request a manual override. The filter may be customized to the needs of the user(s). Next the Modify component handles the task of query modification. This is done to address the uniqueness of search criteria present across individual search engines. The effect of the modifications is to maximize the success rates of searches across the suite of search engines interrogated.

The modified queries are sent to the search engines and the returned results are analyzed, by the Analyze component, to determine structure, content type and viability. At this stage redundant documents are eliminated and common web file types are handled including .HTML, .doc, .pdf, .xml and .ps. Current search engines sometimes produce large volumes of irrelevant results. To tackle this problem we must consider semantic issues, as well as structural and syntactic ones. Standard IR techniques are applied to focus on the issue of relevance in the retrieved documents collection. Many tools exist to aid us in applying both IR and data retrieval techniques to the results obtained from the web (Manning et al, 2007; Zhenyu et al, 2002; Daconta et al, 2003).

Semantic Issues

In the Data Warehouse much attention is paid to retaining the purity, consistency and integrity of data originating from operational databases. These databases take several steps to codify meaning through the use of careful design, data entry procedures, database triggers, etc. The use of explicit meta-data safeguards the meaning of data in these databases. Unfortunately on the web the situation is often chaotic. One promising avenue in addressing the issue of relevance in a heterogeneous environment is the use of formal, knowledge representation constructs known as Ontologies. These constructs have again recently been the subject of revived interest in view of Semantic Web initiatives. In our model we plan to use a domain-specific ontology or taxonomy in the format module to match the results' terms and hence distinguish relevant from non-relevant results (Ding et al, 2007; Decker et al, 2000; Chakrabarti, 2002; Kalfoglou et al, 2004; Hassell et al, 2006; Holzinger et al, 2006).

Figure 2. Meta-data engine operation



Data Synchronization

Data entering the warehouse must be synchronized due to the fact that several sources are utilized. Without synchronization the integrity of the data may be threatened. Data coming from the web should also be synchronized where possible. There is also the issue of the time basis of information including page postings, retrieval times, and page expiration dates, etc. Calculating the time basis of information on the web is an inexact science and can sometimes rely on tangential evidence. Some auxiliary time basis indicators include Internet Archives, Online Libraries, web server logs, content analysis and third party reporting. For instance, content analysis on a date field, in a prominent position relative to a heading, may reveal the publication date.

Analysis of the Model

The model seeks to relieve information overload as users may compose naïve queries which will be augmented and tailored to individual search engines. When results are retrieved they are analyzed to produce the necessary meta-data which allows for the integration of relevant external data into the warehouse.

The strengths of this model include:

- Value-added data
- Flexibility
- Generation of meta-data
- Extensibility
- Security
- Independence (both Logical & Physical)
- Relies on proven technologies

This model, when implemented fully, will extend the usefulness of data in the warehouse by allowing its ageing internal data stores to have much needed context in the form of external web based data. Flexibility is demonstrated since the M-DE is considered a specialized activity under a separate administration and queries are tailored to specific search engines. An important side effect is the generation of meta-data. Relevant descriptors of stored data are as important as the data itself. This meta-data is used to inform the system in relation to future searches.

The logical and physical independence seen in the tri-partite nature of the model allows for optimizations, decreases in development times and enhanced maintainability of the system. Security is of vital concern especially in relation to the Internet. The model bolsters security by providing a buffer between an unpredictable online environment and the data warehouse. We envisage development in at least 3 languages, especially SQL for the warehouse proper, Java for the search engine components and Perl to handle the parsing intensive components of the M-DE.

Some obvious weaknesses of the model include:

- Inexact matching and hence skewed estimations of relevance
- Handling of granularity
- Need to store large volumes of irrelevant information
- Manual fine-tuning required by system administrators
- Handling of Multimedia content
- Does not directly address inaccessible “Deep Web” databases

FUTURE TRENDS

We are already considering the rise of newer modes of external data sources on the web such as blogs and RSS feeds. These may well become more important than the e-zines, online newspapers and electronic forums of today. Search Engine technology is continuing to mature. Heavy investments by commercial engines like Yahoo!, Google and MSN are starting to yield results. We expect that future searches will handle relevance, multimedia and data on the Deep Web with far greater ease. Developments on the Semantic Web, particularly in areas such as Ontological Engineering and Health Care (Eysenbach, 2003; Qazi, 2006; Sheth, 2006), will allow web-based data to be far more transparent to software agents. The data warehouse environment will continue to evolve, having to become more nimble and more accepting of data in diverse formats, including multimedia. The issue of dirty data in the warehouse must be tackled, especially as the volume of data in the warehouse continues to mushroom (Kim, 2003).

CONCLUSION

The model presented seeks to promote the availability and quality of external data for the warehouse through

the introduction of an intermediate data-staging layer. Instead of clumsily seeking to combine the highly structured warehouse data with the lax and unpredictable web data, the meta-data engine we propose mediates between the disparate environments. Key features are the composition of domain specific queries which are further tailor made for individual entries in the suite of search engines being utilized. The ability to disregard irrelevant data through the use of Information Retrieval (IR), Natural Language Processing (NLP) and/or Ontologies is also a plus. Furthermore the exceptional independence and flexibility afforded by our model will allow for rapid advances as niche-specific search engines and more advanced tools for the warehouse become available.

REFERENCES

Bergman, M. (August 2001). The deep Web:Surfacing hidden value. BrightPlanet. *Journal of Electronic Publishing*, 7(1). Retrieved from http://beta.brightplanet.com/deepcontent/tutorials/DeepWeb/index.asp

Berson, A. and Smith, S.J. (1997). *Data Warehousing, Data Mining and Olap*. New York: McGraw-Hill.

Chakrabarti, S. (2002). *Mining the web: Analysis of Hypertext and Semi-Structured Data.* New York: Morgan Kaufman.

Crescenzi, V., Mecca, G., & Merialdo, P. (2001). ROADRUNNER: Towards Automatic Data Extraction from Large Web Sites. Paper presented at the 27th International Conference on Very Large Databases, Rome, Italy.

Daconta, M. C., Obrst, L. J., & Smith, K. T. (2003). The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management: Wiley.

Day, A. (2004). Data Warehouses. *American City & County*. 119(1), 18.

Decker, S., van Harmelen, F., Broekstra, J., Erdmann, M., Fensel, D., Horrocks, I., et al. (2000). The Semantic Web: The Roles of XML and RDF. IEEE Internet Computing, 4(5), 63-74.

Devlin, B. (1998). Meta-data: The Warehouse Atlas. *DB2 Magazine,* 3(1), 8-9.

Ding, Y., Lonsdale, D.W., Embley, D.W., Hepp, M., Xu, L. (2007). Generating Ontologies via Language Components and Ontology Reuse. NLDB 2007: 131-142.

Dumbill, E. (2000). The Semantic Web: A Primer. Retrieved Sept. 2004, 2004, from http://www.xml.com/pub/a/2000/11/01/semanticweb/index.html

Eysenbach, G. (2003). The Semantic Web and healthcare consumers: a new challenge and opportunity on the horizon. Intl J. Healthcare Technology and Management, 5(3/4/5), 194-212.

Hassell, J., Aleman-Meza, B., & Arpinar, I. B. (2006). Ontology-Driven Automatic Entity Disambiguation in Unstructured Text. Paper presented at the ISWC 2006, Athens, GA, USA.

Holzinger, W., Kr¨upl, B., & Herzog, M. (2006). Using Ontologies for Extracting Product Features from Web Pages. Paper presented at the ISWC 2006, Athens, GA, USA.

Imhoff, C., Galemmo, N. and Geiger, J. G. (2003). *Mastering Data Warehouse Design: Relational and Dimensional Techniques*. New York: John Wiley & Sons**.**

Inmon, W.H. (2002). *Building the Data Warehouse, 3rd ed*. New York: John Wiley & Sons.

Kalfoglou, Y., Alani, H., Schorlemmer, M., & Walton, C. (2004). On the emergent Semantic Web and overlooked issues. Paper presented at the 3rd International Semantic Web Conference (ISWC'04).

Kim, W., et al. (2003). "A Taxonomy of Dirty Data". *Data Mining and Knowledge Discovery*, 7, 81-99.

Kimball, R. and Ross, M. (2002). *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling, 2nd ed*. New York: John Wiley & Sons.

Laender, A. H. F., Ribeiro-Neto, B. A., da Silva, A. S., & Teixeira, J. S. (2002). A Brief Survey of Web Data Extraction Tools. SIGMOD Record, 31(2), 84-93.

Ladley, J. (March 2007). "Beyond the Data Warehouse: A Fresh Look". DM Review Online. Available at http://dmreview.com

Manning, C.D, Raghavan, P., Schutze, H. (2007). Introduction to Information Retrieval. Cambridge University Press.

Peter, H. & Greenidge, C. (2005) "Data Warehousing Search Engine". Encyclopedia of Data Warehousing and

Mining, Vol. 1. J. Wang (ed), Idea Group Publishing, ISBN: 1591405572, pp. 328-333.

Qazi, F.A. (2006). Use of Semantic Web in Health Care Systems. SWWS 2006, Las Vegas, Nevada, June 26-29.

Ross, M. (Oct. 2006). "Four Fixes Refurbish Legacy Data Warehouses". Intelligent Enterprise, 9(10), 43-45. Available at http://www.intelligententerprise.com

Shah, U., Finin, T., Joshi, A., Cost, R. S., & Mayfield, J. (2002). Information Retrieval on the Semantic Web. Paper presented at the Tenth International Conference on Information and Knowledge Management (CIKM 2002), McLean, Virginia, USA.

Sheth, A. (2006). Semantic Web applications in Financial Industry, Government, Health Care and Life Sciences. AAAI Spring Symposium on SWEG, Palo Alto, California, March 2006.

Strand, M. & Wangler, B. (June 2004). Incorporating External Data into Data Warehouses – Problem Identified and Contextualized. *Proceedings of the 7th International Conference on Information Fusion,* Stockholm, Sweden, 288-294.

Sullivan, D. (2000). Search Engines Review Chart. [Electronic version], retrieved June 10, 2002, from http://searchenginewatch.com.

Schwartz, E. (2003). Data Warehouses Get Active. *InfoWorld*, 25(48), 12-13.

Wang, J. & Lochovsky, F.H. (2003). Data extraction and label assignment for Web databases. *WWW2003 Conference*, Budapest, Hungary.

Winter, R. & Burns, R. (Nov. 2006). "Climb Every Warehouse". Intelligent Enterprise; 9(11) 31-35. Available at http://www.intelligententerprise.com

Zhenyu, H., Chen, L., Frolick, M. (Winter 2002). "Integrating Web Based Data Into A Data Warehouse". Information Systems Management; 19(1) 23-34.

Zillman, M. P. (2005). *Deep Web research 2005*. Retrieved from http://www.llrx.com/features/deepweb2005.htm

KEY TERMS

Decision Support System (DSS): An interactive arrangement of computerized tools tailored to retrieve and display data regarding business problems and queries.

Deep Web: Denotes those significant but often neglected portions of the web where data is stored in inaccessible formats that cannot be readily indexed by the major search engines. In the literature the term "*Invisible Web*" is sometimes used.

External data: Data originating from other than the operational systems of a corporation.

Metadata: Data about data; in the data warehouse it describes the contents of the data warehouse.

Operational Data: Data used to support the daily processing a company does.

Refresh Cycle: The frequency with which the data warehouse is updated : for example, once a week.

Semantic Web: Area of active research in which XML based technologies are being used to make web data "smarter" so that it can be readily handled by software agents.

Transformation: The conversion of incoming data into the desired form.

Analytical Competition for Managing Customer Relations

Dan Zhu
Iowa State University, USA

INTRODUCTION

With the advent of technology, information is available in abundance on the World Wide Web. In order to have appropriate and useful information users must increasingly use techniques and automated tools to search, extract, filter, analyze and evaluate desired information and resources. Data mining can be defined as the extraction of implicit, previously unknown, and potentially useful information from large databases.

On the other hand, text mining is the process of extracting the information from an unstructured text. A standard text mining approach will involve categorization of text, text clustering, and extraction of concepts, granular taxonomies production, sentiment analysis, document summarization, and modeling (Fan et al, 2006). Furthermore, Web mining is the discovery and analysis of useful information using the World Wide Web (Berry, 2002; Mobasher, 2007). This broad definition encompasses "web content mining," the automated search for resources and retrieval of information from millions of websites and online databases, as well as "web usage mining," the discovery and analysis of users' website navigation and online service access patterns.

Companies are investing significant amounts of time and money on creating, developing, and enhancing individualized customer relationship, a process called customer relationship management or CRM. Based on a report by the Aberdeen Group, worldwide CRM spending reached close to $20 billion by 2006. Today, to improve the customer relationship, most companies collect and refine massive amounts of data available through the customers. To increase the value of current information resources, data mining techniques can be rapidly implemented on existing software and hardware platforms, and integrated with new products and systems (Wang et al., 2008). If implemented on high-performance client/server or parallel processing computers, data mining tools can analyze enormous databases to answer customer-centric questions such as, "Which clients have the highest likelihood of responding to my next promotional mailing, and why." This paper provides a basic introduction to data mining and other related technologies and their applications in CRM.

BACKGROUND

Customer Relationship Management

Customer relationship management (CRM) is an enterprise approach to customer service that uses meaningful communication to understand and influence consumer behavior. The purpose of the process is two-fold: 1) to impact all aspects to the consumer relationship (e.g., improve customer satisfaction, enhance customer loyalty, and increase profitability) and 2) to ensure that employees within an organization are using CRM tools. The need for greater profitability requires an organization to proactively pursue its relationships with customers (Gao et al., 2007). In the corporate world, acquiring, building, and retaining customers are becoming top priorities. For many firms, the quality of its customer relationships provides its competitive edge over other businesses. In addition, the definition of "customer" has been expanded to include immediate consumers, partners and resellers—in other words, virtually everyone who participates, provides information, or requires services from the firm.

Companies worldwide are beginning to realize that surviving an intensively competitive and global marketplace requires closer relationships with customers. In turn, enhanced customer relationships can boost profitability three ways: 1) reducing cots by attracting more suitable customers; 2) generating profits through cross-selling and up-selling activities; and 3) extending profits through customer retention. Slightly expanded explanations of these activities follow.

- **Attracting more suitable customers:** Data mining can help firms understand which customers are most likely to purchase specific products and services, thus enabling businesses to develop targeted marketing programs for higher response rates and better returns on investment.
- **Better cross-selling and up-selling:** Businesses can increase their value proposition by offering additional products and services that are actually desired by customers, thereby raising satisfaction levels and reinforcing purchasing habits.
- **Better retention:** Data mining techniques can identify which customers are more likely to defect and why. This information can be used by a company to generate ideas that allow them maintain these customers.

In general, CRM promises higher returns on investments for businesses by enhancing customer-oriented processes such as sales, marketing, and customer service. Data mining helps companies build personal and profitable customer relationships by identifying and anticipating customer's needs throughout the customer lifecycle.

Data Mining: An Overview

Data mining can help reduce information overload and improve decision making. This is achieved by extracting and refining useful knowledge through a process of searching for relationships and patterns from the extensive data collected by organizations. The extracted information is used to predict, classify, model, and summarize the data being mined. Data mining technologies, such as rule induction, neural networks, genetic algorithms, fuzzy logic and rough sets, are used for classification and pattern recognition in many industries.

Data mining builds models of customer behavior using established statistical and machine learning techniques. The basic objective is to construct a model for one situation in which the answer or output is known, and then apply that model to another situation in which the answer or output is sought. The best applications of the above techniques are integrated with data warehouses and other interactive, flexible business analysis tools. The analytic data warehouse can thus improve business processes across the organization, in areas such as campaign management, new product rollout, and fraud detection. Data mining integrates different technologies to populate, organize, and manage the data store. Since quality data is crucial to accurate results, data mining tools must be able to "clean" the data, making it consistent, uniform, and compatible with the data store. Data mining employs several techniques to extract important information. Operations are the actions that can be performed on accumulated data, including predictive modeling, database segmentation, link analysis, and deviation detection.

Statistical procedures can be used to apply advanced data mining techniques to modeling (Yang & Zhu, 2002; Huang et al., 2006). Improvements in user interfaces and automation techniques make advanced analysis more feasible. There are two groups of modeling and associated tools: theory-driven and data driven. The purpose of theory-driven modeling, also called hypothesis testing, is to substantiate or disprove a priori notions. Thus, theory-driven modeling tools ask the user to specify the model and then test its validity. On the other hand, data-driven modeling tools generate the model automatically based on discovered patterns in the data. The resulting model must be tested and validated prior to acceptance. Since modeling is an evolving and complex process, the final model might require a combination of prior knowledge and new information, yielding a competitive advantage (Davenport & Harris, 2007).

MAIN THRUST

Modern data mining can take advantage of increasing computing power and high-powered analytical techniques to reveal useful relationships in large databases (Han & Kamber, 2006; Wang et al., 2007). For example, in a database containing hundreds of thousands of customers, a data mining process can process separate pieces of information and uncover that 73% of all people who purchased sport utility vehicles also bought outdoor recreation equipment such as boats and snowmobiles within three years of purchasing their SUVs. This kind of information is invaluable to recreation equipment manufacturers. Furthermore, data mining can identify potential customers and facilitate targeted marketing.

CRM software applications can help database marketers automate the process of interacting with their customers (Kracklauer et al., 2004). First, data-

base marketers identify market segments containing customers or prospects with high profit potential. This activity requires processing of massive amounts of data about people and their purchasing behaviors. Data mining applications can help marketers streamline the process by searching for patterns among the different variables that serve as effective predictors of purchasing behaviors. Marketers can then design and implement campaigns that will enhance the buying decisions of a targeted segment, in this case, customers with high income potential. To facilitate this activity, marketers feed the data mining outputs into campaign management software that focuses on the defined market segments. Here are three additional ways in which data mining supports CRM initiatives.

- **Database marketing:** Data mining helps database marketers develop campaigns that are closer to the targeted needs, desires, and attitudes of their customers. If the necessary information resides in a database, data mining can model a wide range of customer activities. The key objective is to identify patterns that are relevant to current business problems. For example, data mining can help answer questions such as "Which customers are most likely to cancel their cable TV service?" and "What is the probability that a customer will spent over $120 from a given store?" Answering these types of questions can boost customer retention and campaign response rates, which ultimately increases sales and returns on investment.
- **Customer acquisition:** The growth strategy of businesses depends heavily on acquiring new customers, which may require finding people who have been unaware of various products and services, who have just entered specific product categories (for example, new parents and the diaper category), or who have purchased from competitors. Although experienced marketers often can select the right set of demographic criteria, the process increases in difficulty with the volume, pattern complexity, and granularity of customer data. Highlighting the challenges of customer segmentation has resulted in an explosive growth in consumer databases. Data mining offers multiple segmentation solutions that could increase the response rate for a customer acquisition campaign. Marketers need to use creativity and experience to tailor new and interesting offers for customers identified through data mining initiatives.
- **Campaign optimization:** Many marketing organizations have a variety of methods to interact with current and prospective customers. The process of optimizing a marketing campaign establishes a mapping between the organization's set of offers and a given set of customers that satisfies the campaign's characteristics and constraints, defines the marketing channels to be used, and specifies the relevant time parameters. Data mining can elevate the effectiveness of campaign optimization processes by modeling customers' channel-specific responses to marketing offers.

Database marketing software enables companies to send customers and prospective customers timely and relevant messages and value propositions. Modern campaign management software also monitors and manages customer communications on multiple channels including direct mail, telemarketing, email, Web, point-of-sale, and customer service. Furthermore, this software can be used to automate and unify diverse marketing campaigns at their various stages of planning, execution, assessment, and refinement. The software can also launch campaigns in response to specific customer behaviors, such as the opening of a new account.

Generally, better business results are obtained when data mining and campaign management work closely together. For example, campaign management software can apply the data mining model's scores to sharpen the definition of targeted customers, thereby raising response rates and campaign effectiveness. Furthermore, data mining may help to resolve the problems that traditional campaign management processes and software typically do not adequately address, such as scheduling, resource assignment, etc. While finding patterns in data is useful, data mining's main contribution is providing relevant information that enables better decision making. In other words, it is a tool that can be used along with other tools (e.g., knowledge, experience, creativity, judgment, etc.) to obtain better results. A data mining system manages the technical details, thus enabling decision-makers to focus on critical business questions such as "Which current customers are likely to be interested in our new product?" and "Which market segment is the best for the launch of our new product?"

FUTURE TRENDS

Data mining is a modern technology that offers competitive firms a method to manage customer information, to retain customers, and to pursue new and hopefully profitable customer relationships. With the emergence new technologies, data mining has been further enhanced and segregated into text mining and web mining.

Text Mining

With the advancement and expansion of data mining there is a large scope and need of an area which can serve the purpose various domains. Fusion of techniques from data mining, language, information process retrieval and visual understanding, created an interdisciplinary field called text mining. Text data mining, referred as, text mining is the process of extracting the information from an unstructured text. In order to obtain high text information, a process of pattern division and trends is done. For an efficient text mining system, the unstructured text is parsed and attached or removed some level of linguistic feature, thus making it structured text. A standard text mining approach will involve categorization of text, text clustering, and extraction of concepts, granular taxonomies production, sentiment analysis, document summarization, and modeling.

Text mining involves a two stage processing of text. In the first step a description of document and its content is done. This process is called categorization process. In the second step, called as classification, the document is divided into descriptive categories and an inter document relationship is established. Of the late, text mining has been useful in many areas, i.e. security applications, software applications, academic applications etc. In the competitive world of business, there is a rush to grab the pie of text mining benefits. With every company focusing on customer relationship management the need for a technique to analyze the customer response in an efficient and effective ways is in demand. This is where text mining fills in the void.

Companies generally concentrate in a smaller quantitative picture of customer response and thus neglecting a broader perspective of CRM. Furthermore, people managing CRM do not put a heed to the day to day communications, suggestions, complaints and praises. This leads to further weaken the CR analysis. With the use of text mining a link to the behavioral data can be obtained which will be an additional asset to the standard numerical analysis. With the involvement of text mining which in itself involves artificial intelligence, machine learning and statistics can be very useful in predicting the future course of customer relationship management.

Web Mining

With the explosion of information we are entering into a flood of information. This information explosion is strongly supported by the internet which by all means has become a universal infrastructure of information (Abbasi & Chen, 2007; Turekten & Sharda, 2007). With the fact that web content is exponentially increasing it is getting difficult day by day to get the appropriate information which is as close as possible to what a user is looking for. Web mining can be used in customizing the web sites based on the contents as well as the user interaction. Types of web mining generally include usage mining, content mining and structure mining.

Data mining, text mining and web mining employ many techniques to extract relevant information from massive data sources so that companies can make better business decisions with regard to their customer relationships. Hence, data, text mining and web mining promote the goals of customer relationship management, which are to initiate, develop, and personalize customer relationships by profiling customers and highlighting segments.

However, data mining presents a number of issues that must be addressed. Data privacy is a trigger-button issue (Atahan & Sarkar, 2007; Zhu et al., 2007). Recently, privacy violations, complaints, and concerns have grown in significance as merchants, companies, and governments continue to accumulate and store large amounts of personal data. There are concerns not only about the collection of personal data, but also the analyses and uses of the data. For example, transactional information is collected from the customer for the processing of a credit card payment, then, without prior notification, the information is used for other purposes (e.g., data mining). This action would violate principles of data privacy. Fueled by the public's concerns about the rising volume of collected data and potent technologies, clashes between data privacy and data mining likely will cause higher levels of scrutiny in the coming years. Legal challenges are quite possible in this regard.

There are other issues facing data mining as well (Olson, 2008). Data inaccuracies can cause analyses, results, and recommendations to veer off-track. Customers' submission of erroneous or false information and data type incompatibilities during the data importation process pose real hazards to data mining's effectiveness. Another risk is that data mining might be easily confused with data warehousing. Companies that build data warehouses without implementing data mining software likely will not reach top productivity nor receive the full benefits. Likewise, cross-selling can be a problem if it violates customers' privacy, breaches their trust, or annoys them with unwanted solicitations. Data mining can help to alleviate the latter issue by aligning marketing programs with targeted customers' interests and needs.

CONCLUSION

Despite the potential issues and impediments, the market for data mining is projected to grow by several billion dollars. Database marketers should understand that some customers are significantly more profitable than others. Data mining can help to identify and target these customers, whose data is "buried" in massive databases, thereby helping to redefine and to reinforce customer relationships.

Data mining tools can predict future trends and behaviors that enable businesses to make proactive, knowledge-based decisions. This is one of the reasons why data mining is also known as knowledge discovery. It is the process of analyzing data from different perspectives, grouping the relationships identified and finally concluding to a set of useful information. This set of information can be further analyzed and utilized by companies to increase revenue, cut costs or a combination of both. With the use of data mining business are finding it easy to answer questions pertaining to business intelligence which were difficult to analyze and conclude before.

ACKNOWLEDGMENT

This research is partially supported by a grant from the Icube and a grant from Center for Information Assurance from Iowa State University.

REFERENCES

Abbasi, A., & Chen, H. (2007). Detecting Fake Escrow Websites using Rich Fraud Cues and Kernel Based Methods. *Workshop on Information Systems (WITS 2007).*

Atahan, P., & Sarkar, S. (2007). Designing Websites to learn user profiles. *Workshop on Information Systems (WITS 2007).*

Davenport, T., & Harris, J.G. (2007). *Competing on analytics*. Harvard Business School Press, Boston, MA.

Fan, W., Wallace, L., Rich, S., & Zhang, Z. (2006). Tapping the power of text mining, *Communications of ACM*, *49*(9), 76-82.

Gao, W. Yang, Z. and O. R. Liu Sheng, (2007). *An interest support based subspace clustering approach to predicting repurchases, workshop on information systems (WITS 2007)*, Montreal, Canada.

Han, J., & Kamber, M. (2006). *Data mining: Concepts and techniques.* 2nd edition San Francisco: Morgan Kaufmann Publishers.

Huang, Z., Zhao, H., Suzuki, Y., & Zhu, D. (2006). Predicting airline choices: A decision support perspective and alternative approaches. *International Conference on Information Systems (ICIS 2006)*, Milwaukee, WI.

Kracklauer, D., Quinn Mills, & Seifert, D. (ed.) (2004). *Collaborative customer relationship management: Taking CRM to the next level*. New York: Springer-Verlag.

Mobasher, B. (2007). The adaptive Web: Methods and strategies of web personalization. In P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), *Lecture Notes in Computer Science*, Vol. 4321, (pp. 90-135), Springer, Berlin-Heidelberg.

Olson, D. (2008). Ethical aspects of web log data mining. *International Journal of Information Technology and Management*, forthcoming.

Turekten, O., & Sharda, R. (2007) Development of a fisheye-based information search processing aid (FISPA) for managing information overload in the Web environment. *Decision Support Systems*, *37*, 415-434.

Wang, J., Hu, X., Hollister, K., & Zhu, D. (2007). A comparison and scenario analysis of leading data mining software. *International Journal of Knowledge Management*, *4*(2), 17-34.

Wang, J., Hu, X., & Zhu, D. (2008), Diminishing downsides of data mining. *International Journal on Business Intelligence and Data Mining*, forthcoming.

Yang, Y., & Zhu, D. (2002). Randomized allocation with nonparametric estimation for a multi-armed bandit problem with covariates. *Annals of Statistics*, *30*, 100-121.

Zhu, D., Li, X., & Wu, S. (2007). Identity disclosure protection: A data reconstruction approach for preserving privacy in data mining. *International Conference on Information Systems (ICIS 2007)*, Montreal, Canada.

KEY TERMS

Application Service Providers: Offer outsourcing solutions that supply, develop, and manage application-specific software and hardware so that customers' internal information technology resources can be freed-up.

Business Intelligence: The type of detailed information that business managers need for analyzing sales trends, customers' purchasing habits, and other key performance metrics in the company.

Classification: Distribute things into classes or categories of the same type, or predict the category of categorical data by building a model based on some predictor variables.

Clustering: Identify groups of items that are similar. The goal is to divide a data set into groups such that records within a group are as homogeneous as possible, and groups are as heterogeneous as possible. When the categories are unspecified, this may be called "unsupervised learning".

Genetic Algorithm: Optimization techniques based on evolutionary concepts that employ processes such as genetic combination, mutation and natural selection in a design.

Online Profiling: The process of collecting and analyzing data from website visits, which can be used to personalize a customer's subsequent experiences on the website.

Rough Sets: A mathematical approach to extract knowledge from imprecise and uncertain data.

Rule Induction: The extraction of valid and useful "if-then-else" type of rules from data based on their statistical significance levels, which are integrated with commercial data warehouse and OLAP platforms.

Web Usage Mining: The analysis of data related to a specific user browser along with the forms submitted by the user during a web transaction.

Analytical Knowledge Warehousing for Business Intelligence

Chun-Che Huang
National Chi Nan University, Taiwan

Tzu-Liang (Bill) Tseng
The University of Texas at El Paso, USA

INTRODUCTION

The Information Technology and Internet techniques are rapidly developing. Interaction between enterprises and customers has dramatically changed. It becomes critical that enterprises are able to perform rapid diagnosis and quickly respond to market change. How to apply business intelligence (BI), manage, and diffuse discovered knowledge efficiently and effectively has attracted much attention (Turban *et al.*, 2007). In this chapter, an "analytical knowledge warehousing" approach is proposed to apply business intelligence, and solve the knowledge management and diffusion issues for decision-making. Analytical knowledge is referred to a set of discovered knowledge, i.e., core of BI, which is extricated from databases, knowledge bases, and other data storage systems through aggregating data analysis techniques and domain experts from business perspective. The solution approach includes conceptual framework of analytical knowledge, analytical knowledge externalization, design and implementation of analytical knowledge warehouse. The methodology has integrated with multi-dimensional analytical techniques to efficiently search analytical knowledge documents. The techniques include static and dynamic domains and solve problems from the technical and management standpoints. The use of analytical knowledge warehouse and multidimensional analysis techniques shows the promising future to apply BI and support decision-making in business.

BACKGROUND

As businesses continue to use computer systems for a growing number of functions, they face the challenge of processing and analyzing huge amounts of data and turning it into profits. In response to this, enterprises are trying to build their business intelligence (BI), which is a set of tools and technologies designed to efficiently extract useful information from oceans of data. Business intelligence which introduces advanced technology into enterprise management (such as data warehouses, OLAP, data mining), not only provides enterprises with the ability to obtain necessary information, but also to turn them into useful knowledge that will improve an enterprises' competitive advantage (Xie *et al.*, 2001). The functions of business intelligence include management of data, analysis of data, support of decision, and excellence of business (Liang *et al.*, 2002). Business intelligence system queries a data source, uses techniques such as online analytical processing and data mining to analyze information in the source, and reports the results of its work (Ortiz, 2002). Business intelligence tools enable organizations to understand their internal and external environment through the systematic acquisition, collation, analysis, interpretation and exploitation of information (Chung *et al.*, 2003). However, the primary challenge of BI is how to represent the sets of knowledge discovered by using advanced technologies, manage, and diffuse them. In most cases, enterprises build knowledge management systems. However, these systems do not consider the dynamic characteristics of knowledge activities (Maier, 2007).

In an enterprise, the structure of knowledge activity, which depicts activities in the knowledge life cycle (Alavi and Leidner, 2001) and potential issues in the process, is dynamic (Figure 1). Two systems are observed in Figure 1. The lower part of Figure 1 shows the "knowledge activity" main system, which projects internal and external changes. The upper part of Figure 1 depicts the system, which starts from requirement of solution approach, and is followed by knowledge sharing, knowledge innovation, knowledge similarity, knowledge externalization and break through knowledge. Furthermore, there are two "feedback" mechanisms

in each system. In Figure 1, the solid line represents the flow and relationship between each knowledge activity. In contrast to the solid line, the dashed line represents the model of barrier of knowledge activity, which often occurs in the real world. Note that the dash line also shows "adjusted" feedback, which brings in an adjusted (opposite) function into the system.

Some business approaches focus on the "enhanced" feedback (solid lines) in order to increase effectiveness of Knowledge Management (KM) and decision-making (Alavi and Leidner, 2001). However, those approaches are merely temporary solutions and in ad hoc manners. Those approaches become dysfunctional eventually. Therefore, a leverage approach (i.e., focusing on improving the adjusted feedbacks, which is represented by the dash lines in Figure 1) is practical and desirable to achieve effectiveness of BI in decision-making.

To model analytical knowledge in an explicit and sharable manner and avoid the ineffectiveness of applying BI in decision-making, it is required to makes clarification of the following issues:

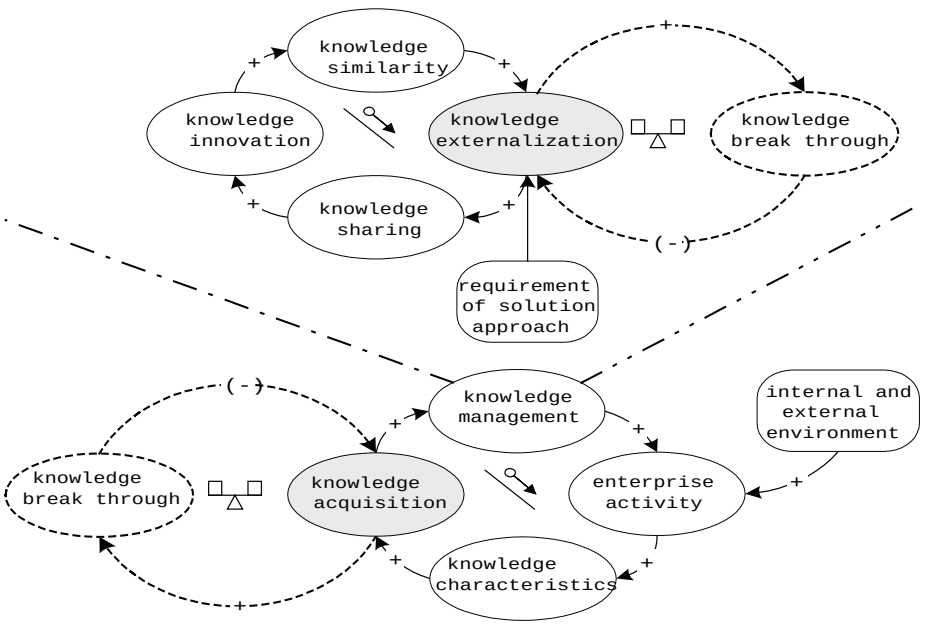
1. Businesses are required to efficiently induce the core knowledge domain (Dieng *et al.*, 1999) and make efforts on high-value and applicable knowledge.
2. From standpoint of technology, knowledge is required to accumulate itself and then be shared with other sources.
3. The lack of well-structured knowledge storage has made the integration of knowledge spiral activities impossible (Nonaka and Takeuchi, 1995).
4. Based on the AK warehousing, the paper uses the multidimensional technique to illustrate the analytical knowledge. The proposed analytical knowledge warehousing eventually stores the paths of analytical knowledge documents, which is classified as non-numerical data.
5. Analytical techniques are used to project potential facts and knowledge in a particular scenario or with some assumptions. The representation is static, rather than dynamic.

MAIN FOCUS

This chapter is based on the data warehousing and knowledge discovery techniques: (1) identify and represent analytical knowledge, which is a result of data analytical techniques, (2) store and manage the analytical knowledge efficiently, (3) accumulate, share, distribute, and integrate the analytical knowledge for BI. This chapter is conceptualized in Figure 2 and illustrated in five levels:

1. In the bottom area, there are two research domains: the left side corresponds to the data storage system

Figure 1. The dynamic structure of knowledge activity in enterprise

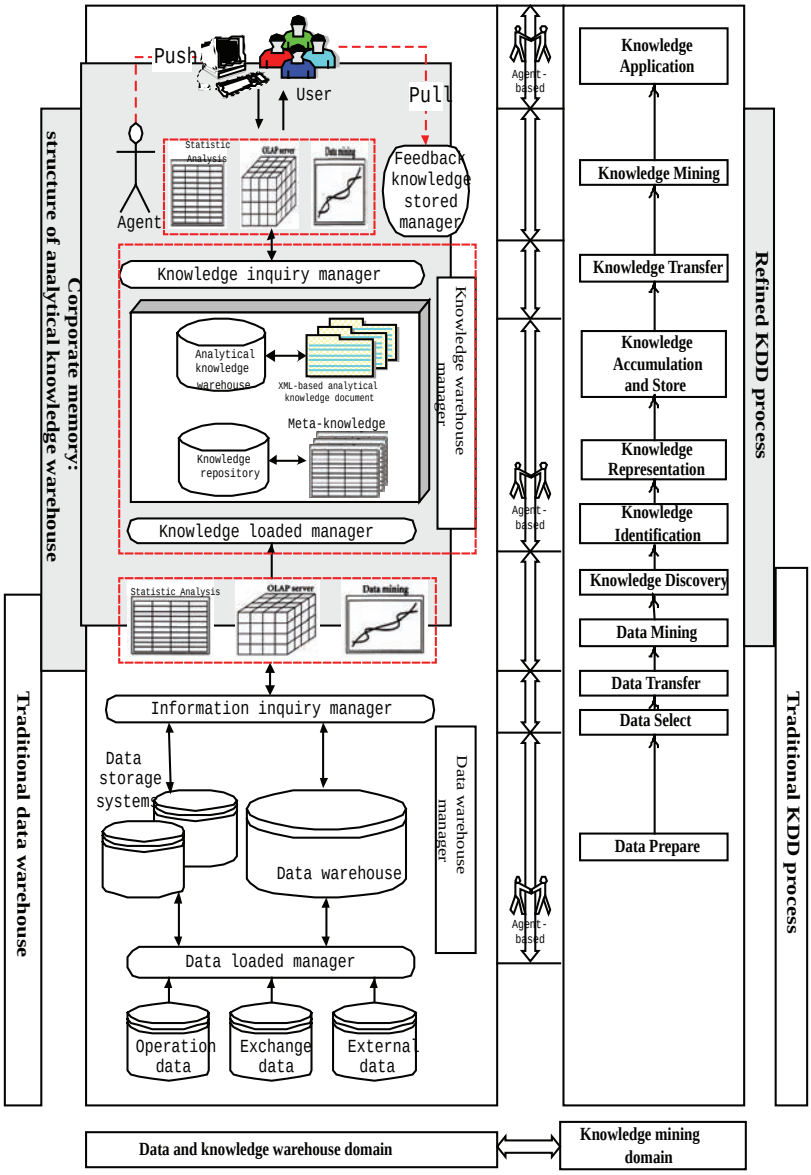


techniques, e.g., data warehousing and knowledge warehousing techniques – a systematic view to aim at the store and management of data and knowledge; and the right side corresponds to the knowledge discovery – an abstract view to aim at the understanding of knowledge discovery processes.

2. The core research of this chapter, the "warehouse for analytical knowledge" is on the left-lower side which corresponds to the traditional data warehouse architecture. Knowledge warehouse of analytical knowledge refers to the management and storage of analytical knowledge documents, which result from the analytical techniques. The structure of AK is defined with the Zachman Framework (Inmon, 2005) and stored in the knowledge depository. The AK warehousing is based on and modified from traditional data warehousing techniques.
3. The white right-bottom side corresponds to the traditional knowledge discovery processes. The right-upper side corresponds to the refined processes of knowledge discovery, which aims at structurally depicting detailed development processes of knowledge discovery.

A

Figure 2. Research structure diagram



4. The two-direction arrow in the middle area refers to the agent-based concept, which aims at the integration of data/knowledge warehousing and knowledge discovery processes and clarification of the mapping between system development and conceptual projection processes. The agent-based systematic approach is proposed to overcome the difficulties of integration, which often occurs in individual and independent development.
5. In the upper area, user knowledge application level, there are two important concepts in the analysis of knowledge warehousing: The "push" concept (the left-upper corner) uses intelligent agents to detect, reason, and respond to the knowledge changes in knowledge warehouse. The agents actively deliver the important messages to knowledge workers to support decision-making on time, rather than in a traditional way, doing jobs by manager's commands. The "pull" concept (the right-upper corner) feedbacks the messages to knowledge warehouse after the users browse and analyze particular analytical knowledge, in such a way to accumulate and share knowledge passively.

Analytical Knowledge (AK)

Analytical knowledge, a core part of BI, is defined as a set of knowledge, which is extricated from databases, knowledge bases, and other data storage systems through aggregating data analysis techniques and domain experts. Data analysis techniques are related to different domains. Each domain comprises several components for example, including statistics, artificial intelligence and database domains. Each domain is not mutually exclusive but correlated to each other. The components under each domain may be reflected at different levels. Those levels are technical application, software platform and fundamental theory. Clearly, all of technical application and software platform are supported by fundamental theory.

Analytical knowledge is different from general knowledge and is constituted only by experts' knowledge, data analysis techniques and data storage systems. Another key feature of analytical knowledge is that domain experts' knowledge should be based on data analysis techniques as well as data storage systems (Davenport and Prusak, 1998).

Numerous classification approaches have been used to distinguish different types of knowledge. For example, knowledge can be further divided into tacit or explicit knowledge (Nonaka, 1991). Analytical knowledge is classified through the following criteria: tacit or explicit, removable or un-removable, expert or un-professional, and strategic resources (Alavi and Leidner, 2001). Note that the definition of expertise includes three categories: know-what, know-how and know-why.

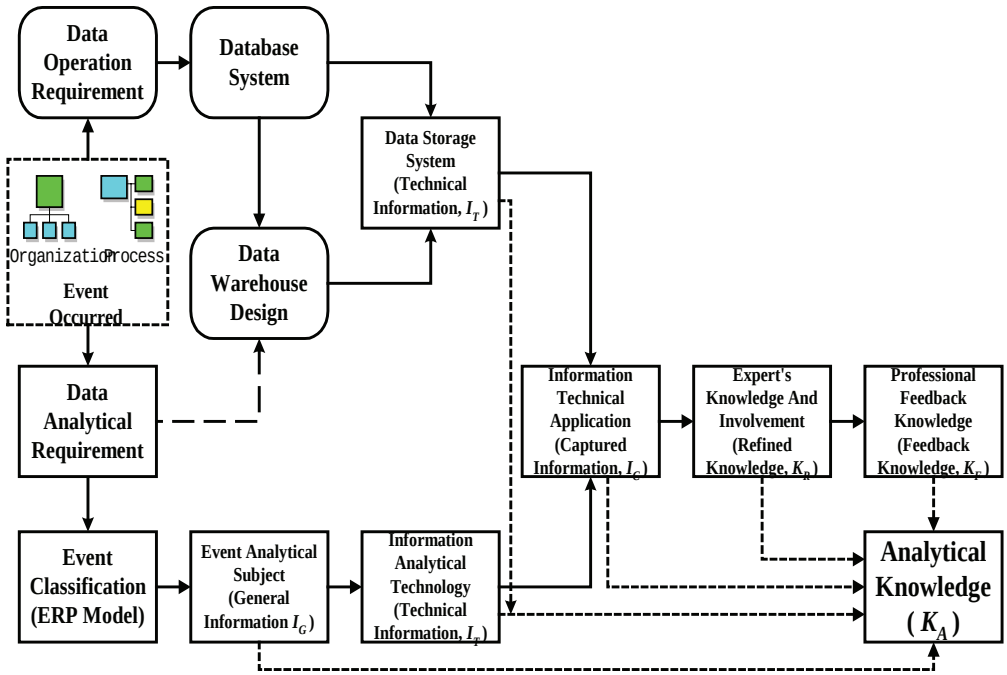
Executive personnel of analytical knowledge are summarized through the Zachman Framework (Inmon *et al.*, 1997). Seven dimensions (entity, location, people, time, motivation, activity, and information) are listed for this summary. Basically, the first six categories correspond to the 5W1H (What, Where, Who, When, Why, and How) technique. Furthermore, five categories are used to represent executive personnel. Data analyzer, knowledge analyzer, knowledge editor, knowledge manager and knowledge user are formed for these categories. Note that the same person might be involved in more than one category since an individual could play multi roles in the organization.

As mentioned before, analytical knowledge is transformed from different stages. Sophisticated process is involved from the first step (e.g., event occurred) to the last step (e.g., knowledge generated). Figure 3 exemplifies the process of generating analytical knowledge. More details are introduced as follows:

(1) A dashed line represents the source of data, which means "event." Here, the needs of data or information analysis from enterprise are triggered by the events from a working process or organization; (2) The blocks represent the entities in the process. There are requirements assessment, events classification, topics identification, techniques identification, techniques application, experts' involvement and group discussions; (3) The oval blocks represent data storage systems; (4) Three different types of arrows are used in the diagram. The solid line arrow illustrates the direction of the process, while the long dash line arrow points out the needs of the data warehouse. The dot dash line shows paths to construct analytical knowledge.

In general, technical information and captured information are highly correlated to the data analysis techniques. Different types of knowledge are produced through different data analysis approaches. For example, an association rule approach in data mining can be implemented to generate technical information

Figure 3. A flow diagram of generating analytical knowledge



while a classification approach can be used to form captured information. However, the nature of these two different kinds of knowledge is significant. Therefore, it is essential to design the structure of knowledge to achieve knowledge externalization and allow the analytical knowledge to reside, e.g., an eXtensible Markup Language (XML) application.

The XML-based externalization process is proposed in Figure 4, which shows the flow chart of the externalization process of analytical knowledge. The results of externalization are stored into a knowledge warehouse by using XML.

To be able to support analytical knowledge warehousing, XML is implemented to define standard documents for externalization, which includes five different categories as above. The XML-based externalization provides a structured and standard way for analytical knowledge representation and development of knowledge warehousing.

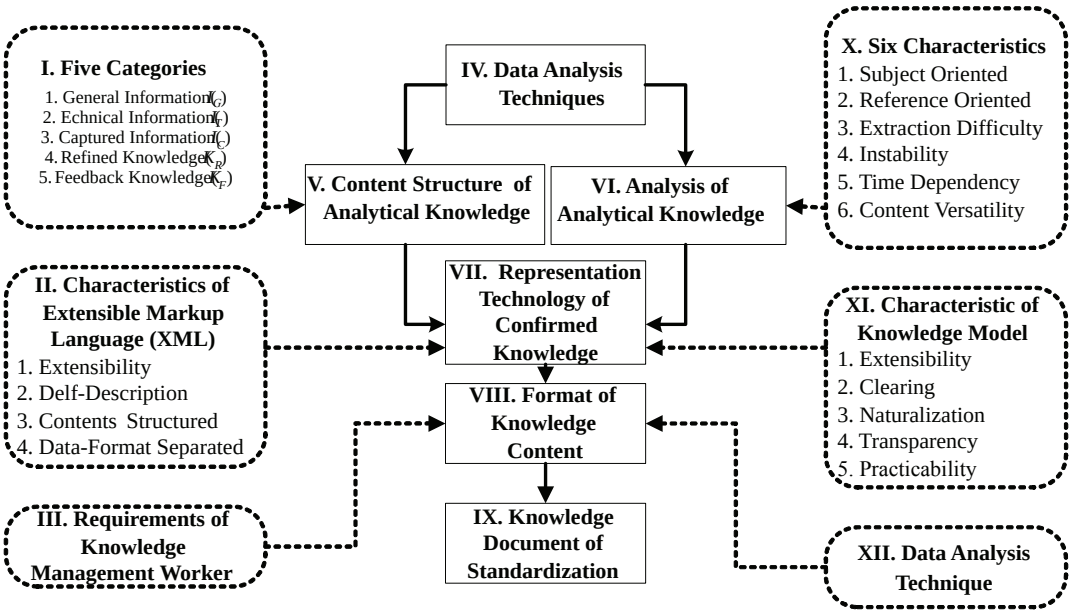
Qualitative Multi-Dimensional Analysis

The multi-dimensional analysis requires numerical data as input entries. This type of quantitative data is critical because it helps the aggregation function to represent the characteristics of data cube and provide rapid access to the entire data warehouse. However, non-numerical data cannot be operated through the multi-dimensional analysis technique. Therefore, this section develops a dynamic solution approach of "qualitative" multidimensional technique for decision-making based on traditional multidimensional techniques and OLAP techniques.

The nature of data classifies two different types of the multi-dimensional analysis. For numerical data, quantitative multi-dimensional analysis (Q-n MA) is used to operate on this type of data. For example, the OLAP fits in this class. For non- numerical data, qualitative multi-dimensional analysis (Q-l MA) is implemented to operate on it. There is not much literature published in the qualitative multi-dimensional analysis area. Contemporary software cannot perform qualitative multi-dimensional analysis effectively due to the lack of commonality of data sharing.

In general, there are two different types of multi-dimensional analysis based on single fact or multiple facts: static and dynamic. The key difference between static and dynamic is that "static" represents merely a fact without taking any variability into consideration while "dynamic" reflects any changes. For example, in the time dimension analysis, emphasizing sales of only one specific month is a "static" status. Modeling the

Figure 4. The flow chart of externalization process of analytical knowledge



sales changes among different months is a "dynamic" analysis. Based on the "static" and "dynamic" status, the operation characteristics of quantitative and qualitative multi-dimensional analysis are as follows:

1. Quantitative static multi-dimensional analysis
   For Q-n MA, the aggregation function embedded in the data cube can be generated. The results of the aggregation function only show the facts under a particular situation. Therefore, it is called "static."
2. Qualitative static multi-dimensional analysis
   The Q-l MA approach operates on non-numerical data. The application of "static" techniques, which shows detailed facts in the data cube, can be observed.
3. Quantitative dynamic multi-dimensional analysis
   In Q-n MA, the dynamic approach should be capable of modeling any input entry changes for the aggregation function from data cubes through a selection of different dimensions and levels. Based on structures of the dimension and level, there are two types of analysis: parallel and vertical analysis. Parallel analysis concentrates on the relationship between elements in the data cube at the same level. Relationship between brothers is an example of this kind of relationship. Vertical analysis emphasizes the relationship in the data cube at different but consecutive levels. In Q-l MA, the dynamic approach includes orthogonal, parallel and vertical analysis.

The orthogonal analysis concentrates on different facts in the same data cube. Because the facts depend on the selection of different dimensions and levels in the cube, the important information could be identified through orthogonal analysis. However the elicited information requires additional analysis through domain experts. Based on dimension and level structures, there are also two different types of analysis: parallel and vertical analysis. Again, "parallel analysis" focuses on the specific level in data cubes and extracts some identical events. Then the events are compared. The "vertical analysis" concentrates on the different but contiguous levels in data cubes. For analytical knowledge, "similarity" or "difference" depends on usage of the data analysis techniques.

Development of Analytical Knowledge Warehouse (AKW)

In this section, analytical knowledge is modeled and stored as a similar way of data warehousing using the

dimensional modeling approach that is one of the best ways to model/store decision support knowledge for data warehouse (Kimball *et al.*, 1998).

The analytical knowledge warehouse stores XML-based analytical knowledge (AK) documents (Li, 2001). Analytical knowledge warehouse emphasizes that "based on data warehouse techniques, the XML-based AK documents can be efficiently and effectively stored and managed." The warehouse helps enterprises carry out the activities in knowledge life cycle and support decision-making in enterprise (Turban and Aronson, 2001).

The purpose of this design is that the knowledge documents can be viewed multi-dimensionally. Dimensional modeling is a logical design technique that seeks to present the data in a standard framework that is intuitive and allows for high-performance access. It is inherently dimensional and adheres to a discipline that uses the relational model with some important restrictions. Every dimensional model is composed of one table with a multipart key, called the fact tables, and a set of smaller tables called dimensional tables. Note that a dimension is a collection of text-like attributes that are highly correlated with each other.

In summary, the goals of analytical knowledge warehouse are to apply the quantitative multi-dimension analysis technique, to explore static and dynamic knowledge, and to support enterprise for decision-making.

FUTURE TRENDS

1. A systematic mechanism to analyze and study the evolution of analytical knowledge over time. For example, mining operations of different types of analytical knowledge evolved could be executed after a particular event (e.g., 911 Event) in order to discover more potential knowledge.
2. Enhancement of maintainability for the well-structured analytical knowledge warehouse. Moreover, revelation of different types of analytical knowledge (e.g., outcomes extracted through the OLAP or AI techniques) is critical and possible solution approaches need to be proposed in further investigation.
3. Application of intelligent agents. Currently, the process of knowledge acquisition is to passively deliver knowledge documents to the user, who requests. If the useful knowledge could be transmitted into the relevant staff members actively, the value of that knowledge can be intensified. It is very difficult to intelligently detect the useful knowledge because the conventional knowledge is not structural. In this research, knowledge has been manipulated through the multi-dimensional analysis technique into analytical knowledge (documents), which is the result of knowledge externalization process. Consequently, if the functionality of intelligent agent can be modeled, then the extracted knowledge can be pushed to desired users automatically. Application of intelligent agent will be a cutting edge approach and the future of application of intelligent agent should be promising.

CONCLUSION

In the chapter, representation of analytical knowledge and system of analytical knowledge warehousing through XML were proposed. Components and structures of the multi-dimensional analysis were introduced. Models of the qualitative multi-dimensional analysis were developed. The proposed approach enhanced the multi-dimensional analysis in terms of apprehending dynamic characteristics and qualitative nature. The qualitative models of the multi-dimensional analysis showed great promise for application in analytical knowledge warehouse. The main contribution of this chapter is that analytical knowledge has been well and structurally defined that could be represented by XML and stored in the AK warehouse. Practically, analytical knowledge not only efficiently facilitates the exploration of useful knowledge but also shows the ability to conduct meaningful knowledge mining on the web for business intelligence.

REFERENCES

Alavi, M. and Leidner, D. E (2001). Review: Knowledge Management and Knowledge Management Systems: Conceptual Foundation and Research Issues, MIS Quarterly. 25 (1) 107-125.

Berson, Alex and Dubov, Larry (2007). Master Data Management and Customer Data Integration for a Global Enterprise. McGraw-Hill, New York.

Chung, Wingyan, Chen, Hsinchun, and Nunamaker, J.F (2003). Business Intelligence Explorer: A Knowledge Map Framework for Discovering Business Intelligence on the web, Proceedings of the 36th Annual Hawaii International Conference on System Sciences.

10 -19.

Davenport, T. H. and Prusak, L (1998). Working Knowledge, Harvard Business School Press, Boston, Massachusetts.

Dieng, R., Corby, O., Giboin, A. and Ribière, M.(1999). Methods and Tools for Corporate Knowledge Management, Journal of Human-Computer Studies. 51 (3) 567-598.

Inmon , W. H. (2005). Building the Data Warehouse, 4/e, John Wiley, New York.

Kimball R., Reeves L., Ross M., Thornrhwaite W (1998). The Data Warehouse Lifecycle Toolkit: Expert Methods for Designing, Developing, and Deploying Data Warehouse, John Wiley & Sons, New York.

Li Ming-Zhong (2001). Development of Analytical Knowledge Warehouse, Dissertation. National Chi-nan University, Taiwan.

Liang, Hao, Gu, Lei, and Wu, Qidi (2002). A Business Intelligence-Based Supply Chain Management Decision Support System, Proceedings of the 4th World Congress on Intelligent Control and Automation. 4 2622-2626.

Maier, Ronald (2007), Knowledge Management Systems: Information and Communication Technologies for Knowledge Management, Springer, New York.

Nonaka I (1991). The Knowledge-creating Company, Harvard Business Review. November-December 69 96-104.

Nonaka I., H. Takeuchi (1995). The Knowledge Creating Company: How Japanese 8Companies Create the Dynamics of Innovation, Oxford University Press, New York.

Ortiz, S., Jr (2002). Is Business Intelligence a Smart Move? Computer. 35(7) 11 -14.

Turban, Efraim, Aronson, Jay E, Liang, Ting-Peng, and Sharda, Ramesh (2007), Decision Support and Business Intelligence Systems (8th Edition), Prentice Hall, New Jersey.

Xie, Wei, Xu, Xiaofei, Sha, Lei, Li, Quanlong, and Liu, Hao (2001), Business Intelligence Based Group Decision Support System, Proceedings of ICII 2001 (Beijing), International Conferences on Info-tech and Info-net. 5 295-300.

KEY TERMS

Business Intelligence: Sets of tools and technologies designed to efficiently extract useful information from oceans of data.

Data Analysis Techniques: The technique to analyze data such as data warehouses, OLAP, data mining.

Data Mining: The nontrivial extraction of implicit, previously unknown, and potentially useful information from data and the science of extracting useful information from large data sets or databases.

Data Warehouses: The main repository of an organization's historical data, its corporate memory. It contains the raw material for management's decision support system. The critical factor leading to the use of a data warehouse is that a data analyst can perform complex queries and analysis, such as data mining, on the information without slowing down the operational systems.

Knowledge Management: Manage knowledge in an efficient way through knowledge externalization, sharing, innovation, and socialization.

On-Line Analytical Processing: An approach to quickly providing answers to analytical queries that are multidimensional in nature.

Qualitative Data: The data extremely varied in nature includes virtually any information that can be captured that is not numerical in nature.

Anomaly Detection for Inferring Social Structure

Lisa Friedland
University of Massachusetts Amherst, USA

INTRODUCTION

In traditional data analysis, data points lie in a Cartesian space, and an analyst asks certain questions: (1) What distribution can I fit to the data? (2) Which points are outliers? (3) Are there distinct clusters or substructure? Today, data mining treats richer and richer types of data. Social networks encode information about people and their communities; relational data sets incorporate multiple types of entities and links; and temporal information describes the dynamics of these systems. With such semantically complex data sets, a greater variety of patterns can be described and views constructed of the data.

This article describes a specific social structure that may be present in such data sources and presents a framework for detecting it. The goal is to identify *tribes*, or small groups of individuals that intentionally coordinate their behavior—individuals with enough in common that they are unlikely to be acting independently.

While this task can only be conceived of in a domain of interacting entities, the solution techniques return to the traditional data analysis questions. In order to find hidden structure (3), we use an anomaly detection approach: develop a model to describe the data (1), then identify outliers (2).

BACKGROUND

This article refers throughout to the case study by Friedland and Jensen (2007) that introduced the tribes task. The National Association of Securities Dealers (NASD) regulates the securities industry in the United States. (Since the time of the study, NASD has been renamed the Financial Industry Regulatory Authority.) NASD monitors over 5000 securities firms, overseeing their approximately 170,000 branch offices and 600,000 employees that sell securities to the public. One of NASD's primary activities is to predict and prevent fraud among these employees, called registered representatives, or *reps*. Equipped with data about the reps' past employments, education, and "disclosable events," it must focus its investigatory resources on those reps most likely to engage in risky behavior. Publications by Neville et al. (2005) and Fast et al. (2007) describe the broader fraud detection problem within this data set.

NASD investigators suspect that fraud risk depends on the social structure among reps and their employers. In particular, some cases of fraud appear to be committed by what we have termed *tribes*—groups of reps that move from job to job together over time. They hypothesized such coordinated movement among jobs could be predictive of future risk. To test this theory, we developed an algorithm to detect tribe behavior. The algorithm takes as input the employment dates of each rep at each branch office, and outputs small groups of reps who have been co-workers to a striking, or anomalous, extent.

This task draws upon several themes from data mining and machine learning:

Inferring latent structure in data. The data we observe may be a poor view of a system's underlying processes. It is often useful to reason about objects or categories we believe exist in real life, but that are not explicitly represented in the data. The hidden structures can be inferred (to the best of our ability) as a means to further analyses, or as an end in themselves. To do this, typically one assumes an underlying model of the full system. Then, a method such as the expectation-maximization algorithm recovers the best match between the observed data and the hypothesized unobserved structures. This type of approach is ubiquitous, appearing for instance in mixture models and clustering (MacKay, 2003), and applied to document and topic models (Hofmann, 1999; Steyvers, et al. 2004).

In relational domains, the latent structure most commonly searched for is clusters. Clusters (in graphs) can be described as groups of nodes densely connected by edges. Relational clustering algorithms hypothesize the existence of this underlying structure, then partition the data so as best to reflect the such groups (Newman, 2004; Kubica et al., 2002; Neville & Jensen, 2005). Such methods have analyzed community structures within, for instance, a dolphin social network (Lusseau & Newman, 2004) and within a company using its network of emails (Tyler et al., 2003). Other variations assume some alternative underlying structure. Gibson et al. (1998) use notions of hubs and authorities to reveal communities on the web, while a recent algorithm by Xu et al. (2007) segments data into three types—clusters, outliers, and hub nodes.

For datasets with links that change over time, a variety of algorithms have been developed to infer structure. Two projects are similar to tribe detection in that they search for specific scenarios of malicious activity, albeit in communication logs: Gerdes et al. (2006) look for evidence of chains of command, while Magdon-Ismail et al. (2003) look for hidden groups sending messages via a public forum.

For the tribes task, the underlying assumption is that most individuals act independently in choosing employments and transferring among jobs, but that certain small groups make their decisions jointly. These tribes consist of members who have worked together unusually much in some way. Identifying these unusual groups is an instance of anomaly detection.

Anomaly detection. Anomalies, or outliers, are examples that do not fit a model. In the literature, the term anomaly detection often refers to intrusion detection systems. Commonly, any deviations from normal computer usage patterns, patterns which are perhaps learned from the data as by Teng and Chen (1990), are viewed as signs of potential attacks or security breaches. More generally for anomaly detection, Eskin (2000) presents a mixture model framework in which, given a model (with unknown parameters) describing normal elements, a data set can be partitioned into normal versus anomalous elements. When the goal is fraud detection, anomaly detection approaches are often effective because, unlike supervised learning, they can highlight both rare patterns plus scenarios not seen in training data. Bolton and Hand (2002) review a number of applications and issues in this area.

MAIN FOCUS

As introduced above, the tribe-detection task begins with the assumption that most individuals make choices individually, but that certain small groups display anomalously coordinated behavior. Such groups leave traces that should allow us to recover them within large data sets, even though the data were not collected with them in mind.

In the problem's most general formulation, the input is a bipartite graph, understood as linking individuals to their affiliations. In place of reps working at branches, the data could take the form of students enrolled in classes, animals and the locations where they are sighted, or customers and the music albums they have rated. A tribe of individuals choosing their affiliations in coordination, in these cases, becomes a group enrolling in the same classes, a mother-child pair that travels together, or friends sharing each other's music. Not every tribe will leave a clear signature, but some groups will have sets of affiliations that are striking, either in that a large number of affiliations are shared, or in that the particular combination of affiliations is unusual.

Framework

We describe the algorithm using the concrete example of the NASD study. Each rep is employed at a series of branch offices of the industry's firms. The basic framework consists of three procedures:

1. For every pair of reps, identify which branches the reps share.
2. Assign a similarity score to each pair of reps, based on the branches they have in common.
3. Group the most similar pairs into tribes.

Step 1 is computationally expensive, but straightforward: For each branch, enumerate the pairs of reps who worked there simultaneously. Then for each pair of reps, compile the list of all branches they shared.

The similarity score of Step 2 depends on the choice of model, discussed in the following section. This is the key component determining what kind of groups the algorithm returns.

After each rep pair is assigned a similarity score, the modeler chooses a threshold, keeps only the most highly similar pairs, and creates a graph by placing an

edge between the nodes of each remaining pair. The graph's connected components become the tribes. That is, a tribe begins with a similar pair of reps, and it expands by including all reps highly similar to those already in the tribe.

Models of "Normal"

The similarity score defines how close two reps are, given the set of branches they share. A pair of reps should be considered close if their set of shared jobs is unusual, i.e., shows signs of coordination. In deciding what makes a set of branches unusual, the scoring function implicitly or explicitly defines a model of normal movement.

Some options for similarity functions include:

Count the jobs. The simplest way to score the likelihood of a given set of branches is to count them: A pair of reps with three branches in common receives the score 3. This score can be seen as stemming from a naïve model of how reps choose employments: At each decision point, a rep either picks a new job, choosing among all branches with equal probability, or else stops working. Under this model, any given sequence of *n* jobs is equally likely and is more likely than a sequence of *n+1* jobs.

Measure duration. Another potential scoring function is to measure how long the pair worked together. This score could arise from the following model: Each day, reps independently choose new jobs (which could be the same as their current jobs). Then, the more days a pair spends as co-workers, the larger the deviation from the model.

Evaluate likelihood according to a Markov process. Each branch can be seen as a state in a Markov process, and a rep's job trajectory can be seen as a sequence generated by this process. At each decision point, a rep either picks a new job, choosing among branches according to the transition probabilities, or else stops working.

This Markov model captures the idea that some job transitions are more common than others. For instance, employees of one firm may regularly transfer to another firm in the same city or the same market. Similarly, when a firm is acquired, the employment data records its workforce as "changing jobs" en masse to the new firm, which makes that job change appear popular. A model that accounts for common versus rare job transitions can judge, for instance, that a pair of independent colleagues in Topeka, Kansas (where the number of firms is limited) is more likely to share three jobs by chance, than a pair in New York City is (where there are more firms to choose from); and that both of these are more likely than for an independent pair to share a job in New York City, then a job in Wisconsin, then a job in Arizona.

The Markov model's parameters can be learned using the whole data set. The likelihood of a particular (ordered) sequence of jobs,

$$P(\text{Branch A} \rightarrow \text{Branch B} \rightarrow \text{Branch C} \rightarrow \text{Branch D})$$

is

$$P(\text{start at Branch A}) \cdot P(A \rightarrow B) \cdot P(B \rightarrow C) \cdot P(C \rightarrow D)$$

The branch-branch transition probabilities and starting probabilities are estimated using the number of reps who worked at each branch and the number that left each branch for the next one. Details of this model, including needed modifications to allow for gaps between shared employments, can be found in the original paper (Friedland & Jensen, 2007).

Use any model that estimates a multivariate binary distribution. In the Markov model above, it is crucial that the jobs be temporally ordered: A rep works at one branch, then another. When the data comes from a domain without temporal information, such as customers owning music albums, an alternative model of "normal" is needed. If each rep's set of branch memberships is represented as a vector of 1's (memberships) and 0's (non-memberships), in a high-dimensional binary space, then the problem becomes estimation of the probability density in this space. Then, to score a particular set of branches shared by a pair of reps, the estimator computes the marginal probability of that set. A number of models, such as Markov random fields, may be suitable; determining which perform well, and which dependencies to model, remains ongoing research.

Evaluation

In the NASD data, the input consisted of the complete table of reps and their branch affiliations, both historical and current. Tribes were inferred using three of the models described above: counting jobs (JOBS),

measuring duration (YEARS), and the Markov process (PROB). Because it was impossible to directly verify the tribe relationships, a number of indirect measures were used to validate the resulting groups, as summarized in Table 1.

The first properties evaluate tribes with respect to their rarity and geographic movement (see table lines 1-2). The remaining properties confirm two joint hypotheses: that the algorithm succeeds at detecting the coordinated behavior of interest, and that this behavior is helpful in predicting fraud. Fraud was measured via a risk score, which described the severity of all reported events and infractions in a rep's work history. If tribes contain many reps known to have committed fraud, then they will be useful in predicting future fraud (line 3). And ideally, groups identified as tribes should fall into two categories. First is high-risk tribes, in which all or most of the members have known infractions. (In fact, an individual with a seemingly clean history in a tribe with several high-risk reps would be a prime candidate for future investigation.) But much more common will be the innocuous tribes, the result of harmless sets of friends recruiting each other from job to job. Within ideal tribes, reps are not necessarily high-risk, but they should match each other's risk scores (line 4).

Throughout the evaluations, the JOBS and PROB models performed well, whereas the YEARS model did not. JOBS and PROB selected different sets of tribes, but the tribes were fairly comparable under most evaluation measures: compared to random groups of reps, tribes had rare combinations of jobs, traveled geographically (particularly for PROB), had higher risk scores, and were homogenous. The tribes identified by YEARS poorly matched the desired properties: not only did these reps not commit fraud, but the tribes often consisted of large crowds of people who shared very typical job trajectories.

Informally, JOBS and PROB chose tribes that differed in ways one would expect. JOBS selected some tribes that shared six or more jobs but whose reps appeared to be caught up in a series of firm acquisitions: many other reps also had those same jobs. PROB selected some tribes that shared only three jobs, yet clearly stood out: Of thousands of colleagues at each branch, only this pair had made any of the job transitions in the series. One explanation why PROB did not perform conclusively better is its weakness at small branches. If a pair of reps works together at a two-person branch, then transfers elsewhere together, the model judges this transfer to be utterly unremarkable, because it is what 100% of their colleagues at that branch (i.e., just the two of them) did. For reasons like this, the model seems to miss potential tribes that work at multiple small branches together. Correcting for this situation, and understanding other such effects, remain as future work.

FUTURE TRENDS

One future direction is to explore the utility of the tribe structure to other domains. For instance, an online bookstore could use the tribes algorithm to infer book clubs—individuals that order the same books at the same times. More generally, customers with unusually similar tastes might want to be introduced; the similarity scores could become a basis for matchmaking on

Table 1. Desirable properties of tribes

Property	Why this is desirable
Tribes share rare combinations of jobs.	An ideal tribe should be fairly unique in its job-hopping behavior.
Tribes are more likely to traverse multiple zip codes.	Groups that travel long distances together are unlikely to be doing so by chance.
Tribes have much higher risk scores than average.	If fraud does tend to occur in tribe-like structures, then on average, reps in tribes should have worse histories.
Tribes are homogenous: reps in a tribe have similar risk scores.	Each tribe should either be innocuous or high-risk.

dating websites, or for connecting researchers who read or publish similar papers. In animal biology, there is a closely related problem of determining family ties, based on which animals repeatedly appear together in herds (Cairns & Schwager, 1987). These "association patterns" might benefit from being formulated as tribes, or even vice versa.

Work to evaluate other choices of scoring models, particularly those that can describe affiliation patterns in non-temporal domains, is ongoing. Additional research will expand our understanding of tribe detection by examining performance across different domains and by comparing properties of the different models, such as tractability and simplicity.

CONCLUSION

The domains discussed here (stock brokers, online customers, etc.) are rich in that they report the interactions of multiple entity types over time. They embed signatures of countless not-yet-formulated behaviors in addition to those demonstrated by tribes.

The tribes framework may serve as a guide to detecting any new behavior that a modeler describes. Key aspects of this approach include searching for occurrences of the pattern, developing a model to describe "normal" or chance occurrences, and marking outliers as entities of interest.

The compelling motivation behind identifying tribes or similar patterns is in detecting hidden, but very real, relationships. For the most part, individuals in large data sets appear to behave independently, subject to forces that affect everyone in their community. However, in certain cases, there is enough information to rule out independence and to highlight coordinated behavior.

REFERENCES

Bolton, R. & Hand, D. (2002). Statistical fraud detection: A review. *Statistical Science*, 17(3), 235-255.

Cairns, S. J. & Schwager, S. J. (1987). A comparison of association indices. *Animal Behaviour*, 35(5), 1454-1469.

Eskin, E. (2000). Anomaly detection over noisy data using learned probability distributions. In *Proc. 17th International Conf. on Machine Learning* (pp. 255-262).

Fast, A., Friedland, L., Maier, M., Taylor, B., & Jensen, D. (2007). Data pre-processing for improved detection of securities fraud in relational domains. In *Proc. 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 941-949).

Friedland, L. & Jensen, D. (2007). Finding tribes: Identifying close-knit individuals from employment patterns. In *Proc. 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 290-299).

Gibson, D., Kleinberg, J., & Raghavan, P. (1998). Inferring Web communities from link topology. In *Proc. 9th ACM Conference on Hypertext and Hypermedia* (pp. 225-234).

Gerdes, D., Glymour, C., & Ramsey, J. (2006). Who's calling? Deriving organization structure from communication records. In A. Kott (Ed.), *Information Warfare and Organizational Structure*. Artech House.

Hofmann, T. (1999). Probabilistic latent semantic analysis. In *Proc. 15th Conference on Uncertainty in AI* (pp. 289-296).

Kubica, J., Moore, A., Schneider, J., & Yang, Y. (2002). Stochastic link and group detection. In *Proc. 18th Nat. Conf. on Artificial Intelligence* (pp. 798-804).

Lusseau, D. & Newman, M. (2004). Identifying the role that individual animals play in their social network. *Proc. R. Soc. London B (Suppl.)* 271, S477-S481.

MacKay, D. (2003). *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press.

Magdon-Ismail, M., Goldberg, M., Wallace, W., & Siebecker, D. (2003). Locating hidden groups in communication networks using hidden Markov models. In *Proc. NSF/NIJ Symposium on Intelligence and Security Informatics* (pp.126-137).

Neville, J. & Jensen, D. (2005). Leveraging relational autocorrelation with latent group models. In *Proc. 5th IEEE Int. Conf. on Data Mining* (pp. 322-329).

Neville, J., Şimşek, Ö., Jensen, D., Komoroske, J., Palmer, K., & Goldberg, H. (2005). Using relational knowledge discovery to prevent securities fraud. In *Proc. 11th ACM Int. Conf. on Knowledge Discovery and Data Mining* (pp. 449-458).

Newman, M. (2004). Fast algorithm for detecting community structure in networks. *Phys. Rev. E* 69, 066133.

Steyvers, M., Smyth, P., Rosen-Zvi, M., & Griffiths, T. (2004). Probabilistic author-topic models for information discovery. In *Proc. 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 306-315).

Teng, H. S. & Chen, K. (1990) Adaptive real-time anomaly detection using inductively generated sequential patterns. In *Proc. IEEE Symposium on Security and Privacy*, (pp. 278-284).

Tyler, J. R., Wilkinson, D. M., & Huberman, B. A. (2003). Email as spectroscopy: Automated discovery of community structure within organizations. In *Communities and Technologies* (pp. 81-96).

Xu, X., Yuruk, N., Feng, Z., & Schweiger, T. (2007). SCAN: A structural clustering algorithm for networks. In *Proc. 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 824-833).

KEY TERMS

Anomaly Detection: Discovering anomalies, or outliers, in data.

Branch: In the NASD schema, a branch is the smallest organizational unit recorded: every firm has one or more branch offices.

Branch Transition: The NASD study examined patterns of job changes. If employees who work at Branch A often work at Branch B next, we say the (branch) transition between Branches A and B is common.

Latent Structure: In data, a structure or pattern that is not explicit. Recovering such structures can make data more understandable, and can be a first step in further analyses.

Markov Process: Model that stochastically chooses a sequence of states. The probability of selecting any state depends only on the previous state.

Registered Representative (rep): Term for individual in the NASD data.

Tribe: Small group of individuals acting in a coordinated manner, e.g., moving from job to job together.

The Application of Data-Mining to Recommender Systems

J. Ben Schafer
University of Northern Iowa, USA

INTRODUCTION

In a world where the number of choices can be overwhelming, recommender systems help users find and evaluate items of interest. They connect users with items to "consume" (purchase, view, listen to, etc.) by associating the content of recommended items or the opinions of other individuals with the consuming user's actions or opinions. Such systems have become powerful tools in domains from electronic commerce to digital libraries and knowledge management. For example, a consumer of just about any major online retailer who expresses an interest in an item – either through viewing a product description or by placing the item in his "shopping cart" – will likely receive recommendations for additional products. These products can be recommended based on the top overall sellers on a site, on the demographics of the consumer, or on an analysis of the past buying behavior of the consumer as a prediction for future buying behavior. This paper will address the technology used to generate recommendations, focusing on the application of data mining techniques.

BACKGROUND

Many different algorithmic approaches have been applied to the basic problem of making accurate and efficient recommender systems. The earliest "recommender systems" were content filtering systems designed to fight information overload in textual domains. These were often based on traditional information filtering and information retrieval systems. Recommender systems that incorporate information retrieval methods are frequently used to satisfy ephemeral needs (short-lived, often one-time needs) from relatively static databases. For example, requesting a recommendation for a book preparing a sibling for a new child in the family. Conversely, recommender systems that incorporate information-filtering methods are frequently used to satisfy persistent information (long-lived, often frequent, and specific) needs from relatively stable databases in domains with a rapid turnover or frequent additions. For example, recommending AP stories to a user concerning the latest news regarding a senator's re-election campaign.

Without computers, a person often receives recommendations by listening to what people around him have to say. If many people in the office state that they enjoyed a particular movie, or if someone he tends to agree with suggests a given book, then he may treat these as recommendations. Collaborative filtering (CF) is an attempt to facilitate this process of "word of mouth." The simplest of CF systems provide generalized recommendations by aggregating the evaluations of the community at large. More personalized systems (Resnick & Varian, 1997) employ techniques such as user-to-user correlations or a nearest-neighbor algorithm.

The application of user-to-user correlations derives from statistics, where correlations between variables are used to measure the usefulness of a model. In recommender systems correlations are used to measure the extent of agreement between two users (Breese, Heckerman, & Kadie, 1998) and used to identify users whose ratings will contain high predictive value for a given user. Care must be taken, however, to identify correlations that are actually helpful. Users who have only one or two rated items in common should not be treated as strongly correlated. Herlocker et al. (1999) improved system accuracy by applying a significance weight to the correlation based on the number of co-rated items.

Nearest-neighbor algorithms compute the distance between users based on their preference history. Distances vary greatly based on domain, number of users, number of recommended items, and degree of co-rating between users. Predictions of how much a user will like an item are computed by taking the weighted average

of the opinions of a set of neighbors for that item. As applied in recommender systems, neighbors are often generated online on a query-by-query basis rather than through the off-line construction of a more thorough model. As such, they have the advantage of being able to rapidly incorporate the most up-to-date information, but the search for neighbors is slow in large databases. Practical algorithms use heuristics to search for good neighbors and may use opportunistic sampling when faced with large populations.

Both nearest-neighbor and correlation-based recommenders provide a high level of personalization in their recommendations, and most early systems using these techniques showed promising accuracy rates. As such, CF-based systems have continued to be popular in recommender applications and have provided the benchmarks upon which more recent applications have been compared.

DATA MINING IN RECOMMENDER APPLICATIONS

The term data mining refers to a broad spectrum of mathematical modeling techniques and software tools that are used to find patterns in data and user these to build models. In this context of recommender applications, the term data mining is used to describe the collection of analysis techniques used to infer recommendation rules or build recommendation models from large data sets. Recommender systems that incorporate data mining techniques make their recommendations using knowledge learned from the actions and attributes of users. These systems are often based on the development of user profiles that can be persistent (based on demographic or item "consumption" history data), ephemeral (based on the actions during the current session), or both. These algorithms include clustering, classification techniques, the generation of association rules, and the production of similarity graphs through techniques such as Horting.

Clustering techniques work by identifying groups of consumers who appear to have similar preferences. Once the clusters are created, averaging the opinions of the other consumers in her cluster can be used to make predictions for an individual. Some clustering techniques represent each user with partial participation in several clusters. The prediction is then an average across the clusters, weighted by degree of participation. Clustering techniques usually produce less-personal recommendations than other methods, and in some cases, the clusters have worse accuracy than CF-based algorithms (Breese, Heckerman, & Kadie, 1998). Once the clustering is complete, however, performance can be very good, since the size of the group that must be analyzed is much smaller. Clustering techniques can also be applied as a "first step" for shrinking the candidate set in a CF-based algorithm or for distributing neighbor computations across several recommender engines. While dividing the population into clusters may hurt the accuracy of recommendations to users near the fringes of their assigned cluster, pre-clustering may be a worthwhile trade-off between accuracy and throughput.

Classifiers are general computational models for assigning a category to an input. The inputs may be vectors of features for the items being classified or data about relationships among the items. The category is a domain-specific classification such as malignant/benign for tumor classification, approve/reject for credit requests, or intruder/authorized for security checks. One way to build a recommender system using a classifier is to use information about a product and a customer as the input, and to have the output category represent how strongly to recommend the product to the customer. Classifiers may be implemented using many different machine-learning strategies including rule induction, neural networks, and Bayesian networks. In each case, the classifier is trained using a training set in which ground truth classifications are available. It can then be applied to classify new items for which the ground truths are not available. If subsequent ground truths become available, the classifier may be retrained over time.

For example, Bayesian networks create a model based on a training set with a decision tree at each node and edges representing user information. The model can be built off-line over a matter of hours or days. The resulting model is very small, very fast, and essentially as accurate as CF methods (Breese, Heckerman, & Kadie, 1998). Bayesian networks may prove practical for environments in which knowledge of consumer preferences changes slowly with respect to the time needed to build the model but are not suitable for environments in which consumer preference models must be updated rapidly or frequently.

Classifiers have been quite successful in a variety of domains ranging from the identification of fraud

and credit risks in financial transactions to medical diagnosis to intrusion detection. Good et al. (1999) implemented induction-learned feature-vector classification of movies and compared the classification with CF recommendations; this study found that the classifiers did not perform as well as CF, but that combining the two added value over CF alone.

One of the best-known examples of data mining in recommender systems is the discovery of association rules, or item-to-item correlations (Sarwar et. al., 2001). These techniques identify items frequently found in "association" with items in which a user has expressed interest. Association may be based on co-purchase data, preference by common users, or other measures. In its simplest implementation, item-to-item correlation can be used to identify "matching items" for a single item, such as other clothing items that are commonly purchased with a pair of pants. More powerful systems match an entire set of items, such as those in a customer's shopping cart, to identify appropriate items to recommend. These rules can also help a merchandiser arrange products so that, for example, a consumer purchasing a child's handheld video game sees batteries nearby. More sophisticated temporal data mining may suggest that a consumer who buys the video game today is likely to buy a pair of earplugs in the next month.

Item-to-item correlation recommender applications usually use current interest rather than long-term customer history, which makes them particularly well suited for ephemeral needs such as recommending gifts or locating documents on a topic of short lived interest. A user merely needs to identify one or more "starter" items to elicit recommendations tailored to the present rather than the past.

Association rules have been used for many years in merchandising, both to analyze patterns of preference across products, and to recommend products to consumers based on other products they have selected. An association rule expresses the relationship that one product is often purchased along with other products. The number of possible association rules grows exponentially with the number of products in a rule, but constraints on confidence and support, combined with algorithms that build association rules with itemsets of n items from rules with n-1 item itemsets, reduce the effective search space. Association rules can form a very compact representation of preference data that may improve efficiency of storage as well as performance. They are more commonly used for larger populations rather than for individual consumers, and they, like other learning methods that first build and then apply models, are less suitable for applications where knowledge of preferences changes rapidly. Association rules have been particularly successfully in broad applications such as shelf layout in retail stores. By contrast, recommender systems based on CF techniques are easier to implement for personal recommendation in a domain where consumer opinions are frequently added, such as online retail.

In addition to use in commerce, association rules have become powerful tools in recommendation applications in the domain of knowledge management. Such systems attempt to predict which Web page or document can be most useful to a user. As Géry (2003) writes, "The problem of finding Web pages visited together is similar to finding associations among itemsets in transaction databases. Once transactions have been identified, each of them could represent a basket, and each web resource an item." Systems built on this approach have been demonstrated to produce both high accuracy and precision in the coverage of documents recommended (Geyer-Schultz et al., 2002).

Horting is a graph-based technique in which nodes are users, and edges between nodes indicate degree of similarity between two users (Wolf et al., 1999). Predictions are produced by walking the graph to nearby nodes and combining the opinions of the nearby users. Horting differs from collaborative filtering as the graph may be walked through other consumers who have not rated the product in question, thus exploring transitive relationships that traditional CF algorithms do not consider. In one study using synthetic data, Horting produced better predictions than a CF-based algorithm (Wolf et al., 1999).

FUTURE TRENDS

As data mining algorithms have been tested and validated in their application to recommender systems, a variety of promising applications have evolved. In this section we will consider three of these applications – meta-recommenders, social data mining systems, and temporal systems that recommend when rather than what.

Meta-recommenders are systems that allow users to personalize the merging of recommendations from

a variety of recommendation sources employing any number of recommendation techniques. In doing so, these systems let users take advantage of the strengths of each different recommendation method. The SmartPad supermarket product recommender system (Lawrence et al., 2001) suggests new or previously unpurchased products to shoppers creating shopping lists on a personal digital assistant (PDA). The SmartPad system considers a consumer's purchases across a store's product taxonomy. Recommendations of product subclasses are based upon a combination of class and subclass associations drawn from information filtering and co-purchase rules drawn from data mining. Product rankings within a product subclass are based upon the products' sales rankings within the user's consumer cluster, a less personalized variation of collaborative filtering. MetaLens (Schafer et al., 2002) allows users to blend content requirements with personality profiles to allow users to determine which movie they should see. It does so by merging more persistent and personalized recommendations, with ephemeral content needs such as the lack of offensive content or the need to be home by a certain time. More importantly, it allows the user to customize the process by weighting the importance of each individual recommendation.

While a traditional CF-based recommender typically requires users to provide explicit feedback, a social data mining system attempts to mine the social activity records of a community of users to implicitly extract the importance of individuals and documents. Such activity may include Usenet messages, system usage history, citations, or hyperlinks. TopicShop (Amento et al., 2003) is an information workspace which allows groups of common Web sites to be explored, organized into user defined collections, manipulated to extract and order common features, and annotated by one or more users. These actions on their own may not be of large interest, but the collection of these actions can be mined by TopicShop and redistributed to other users to suggest sites of general and personal interest. Agrawal et al. (2003) explored the threads of newsgroups to identify the relationships between community members. Interestingly, they concluded that due to the nature of newsgroup postings – users are more likely to respond to those with whom they disagree – "links" between users are more likely to suggest that users should be placed in differing partitions rather than the same partition. Although this technique has not been directly applied to the construction of recommendations, such an application seems a logical field of future study.

Although traditional recommenders suggest what item a user should consume they have tended to ignore changes over time. Temporal recommenders apply data mining techniques to suggest when a recommendation should be made or when a user should consume an item. Adomavicius and Tuzhilin (2001) suggest the construction of a recommendation warehouse, which stores ratings in a hypercube. This multidimensional structure can store data on not only the traditional user and item axes, but also for additional profile dimensions such as time. Through this approach, queries can be expanded from the traditional "what items should we suggest to user X" to "at what times would user X be most receptive to recommendations for product Y." Hamlet (Etzioni et al., 2003) is designed to minimize the purchase price of airplane tickets. Hamlet combines the results from time series analysis, Q-learning, and the Ripper algorithm to create a multi-strategy data-mining algorithm. By watching for trends in airline pricing and suggesting when a ticket should be purchased, Hamlet was able to save the average user 23.8% when savings was possible.

CONCLUSION

Recommender systems have emerged as powerful tools for helping users find and evaluate items of interest. These systems use a variety of techniques to help users identify the items that best fit their tastes or needs. While popular CF-based algorithms continue to produce meaningful, personalized results in a variety of domains, data mining techniques are increasingly being used in both hybrid systems, to improve recommendations in previously successful applications, and in stand-alone recommenders, to produce accurate recommendations in previously challenging domains. The use of data mining algorithms has also changed the types of recommendations as applications move from recommending what to consume to also recommending when to consume. While recommender systems may have started as largely a passing novelty, they clearly appear to have moved into a real and powerful tool in a variety of applications, and that data mining algorithms can be and will continue to be an important part of the recommendation process.

REFERENCES

Adomavicius, G., & Tuzhilin, A. (2001). Extending recommender systems: A multidimensional approach. *IJCAI-01 Workshop on Intelligent Techniques for Web Personalization* (ITWP'2001), Seattle, Washington.

Agrawal, R., Rajagopalan, S., Srikant, R., & Xu, Y. (2003). Mining newsgroups using networks arising from social behavior. In *Proceedings of the Twelfth World Wide Web Conference* (WWW12) (pp. 529-535), Budapest, Hungary.

Amento, B., Terveen, L., Hill, W., Hix, D., & Schulman, R. (2003). Experiments in social data mining: The TopicShop System. *ACM Transactions on Computer-Human Interaction*, *10* (1), 54-85.

Breese, J., Heckerman, D., & Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence* (UAI-98) (pp. 43-52), Madison, Wisconsin.

Etzioni, O., Knoblock, C.A., Tuchinda, R., & Yates, A. (2003). To buy or not to buy: Mining airfare data to minimize ticket purchase price. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 119-128), Washington. D.C.

Géry, M., & Haddad, H. (2003). Evaluation of Web usage mining approaches for user's next request prediction. In *Fifth International Workshop on Web Information and Data Management* (pp. 74-81), Madison, Wisconsin.

Geyer-Schulz, A., & Hahsler, M. (2002). Evaluation of recommender algorithms for an Internet information broker based on simple association rules and on the repeat-buying theory. In *Fourth WEBKDD Workshop: Web Mining for Usage Patterns & User Profiles* (pp. 100-114), Edmonton, Alberta, Canada.

Good, N. et al. (1999). Combining collaborative filtering with personal agents for better recommendations. In *Proceedings of Sixteenth National Conference on Artificial Intelligence* (AAAI-99) (pp. 439-446), Orlando, Florida.

Herlocker, J., Konstan, J.A., Borchers, A., & Riedl, J. (1999). An algorithmic framework for performing collaborative filtering. In *Proceedings of the 1999 Conference on Research and Development in Information Retrieval*, (pp. 230-237), Berkeley, California.

Lawrence, R.D. et al. (2001). Personalization of supermarket product recommendations. *Data Mining and Knowledge Discovery*, *5*(1/2), 11-32.

Lin, W., Alvarez, S.A., & Ruiz, C. (2002). Efficient adaptive-support association rule mining for recommender systems. *Data Mining and Knowledge Discovery*, *6*(1) 83-105.

Resnick, P., & Varian, H.R. (1997). *Communications of the Association of Computing Machinery Special issue on Recommender Systems*, *40*(3), 56-89.

Sarwar, B., Karypis, G., Konstan, J.A., & Reidl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the Tenth International Conference on World Wide Web* (pp. 285-295), Hong Kong.

Schafer, J.B., Konstan, J.A., & Riedl, J. (2001). E-Commerce Recommendation Applications. *Data Mining and Knowledge Discovery*, 5(1/2), 115-153.

Schafer, J.B., Konstan, J.A., & Riedl, J. (2002). Meta-recommendation systems: User-controlled integration of diverse recommendations. In *Proceedings of the Eleventh Conference on Information and Knowledge* (CIKM-02) (pp. 196-203), McLean, Virginia.

Shoemaker, C., & Ruiz, C. (2003). Association rule mining algorithms for set-valued data. *Lecture Notes in Computer Science, 2690*, 669-676.

Wolf, J., Aggarwal, C., Wu, K-L., & Yu, P. (1999). Horting hatches an egg: A new graph-theoretic approach to collaborative filtering. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 201-212), San Diego, CA.

KEY TERMS

Association Rules: Used to associate items in a database sharing some relationship (e.g., co-purchase information). Often takes the for "if this, then that," such as, "If the customer buys a handheld videogame then the customer is likely to purchase batteries."

Collaborative Filtering: Selecting content based on the preferences of people with similar interests.

Meta-Recommenders: Provide users with personalized control over the generation of a single recommendation list formed from the combination of rich recommendation data from multiple information sources and recommendation techniques.

Nearest-Neighbor Algorithm: A recommendation algorithm that calculates the distance between users based on the degree of correlations between scores in the users' preference histories. Predictions of how much a user will like an item are computed by taking the weighted average of the opinions of a set of nearest neighbors for that item.

Recommender Systems: Any system that provides a recommendation, prediction, opinion, or user-configured list of items that assists the user in evaluating items.

Social Data-Mining: Analysis and redistribution of information from records of social activity such as newsgroup postings, hyperlinks, or system usage history.

Temporal Recommenders: Recommenders that incorporate time into the recommendation process. Time can be either an input to the recommendation function, or the output of the function.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 44-48, copyright 2005 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

Applications of Kernel Methods

Gustavo Camps-Valls
Universitat de València, Spain

Manel Martínez-Ramón
Universidad Carlos III de Madrid, Spain

José Luis Rojo-Álvarez
Universidad Rey Juan Carlos, Spain

INTRODUCTION

In this chapter, we give a survey of applications of the kernel methods introduced in the previous chapter. We focus on different application domains that are particularly active in both direct application of well-known kernel methods, and in new algorithmic developments suited to a particular problem. In particular, we consider the following application fields: biomedical engineering (comprising both biological signal processing and bioinformatics), communications, signal, speech and image processing.

KERNEL METHODS IN BIOMEDICINE AND BIOINFORMATICS

Kernel methods have been extensively used to solve biomedical problems. For instance, a study of prediction of cyclosporine dosage in patients after kidney transplantation using neural networks and kernel-based methods was carried out in (Camps-Valls et al., 2002). Recently, (Osowski, Hoai, & Markiewicz, 2004) proposed a committee of experts formed by several support vector machines (SVM) for the recognition of 13 heart rhythm types.

The most impressive results of kernels have been obtained in genomics and computational biology, due to both the special characteristics of data and the great interest in solving biological problems since the Human genome sequencing. Their ability to work with high dimensional data, to process and efficiently integrate non-vectorial string data, make them very suitable to solve various problems arising in computational biology. Since the early papers using SVM in bioinformatics (Mukherjee et al., 1998), the applications of these methods have grown exponentially, and many novel and powerful methods have been developed (only in 2004, more than 1000 papers have been devoted to this topic). The use of kernel methods in computational biology has been accompanied by new developments to match the specificities and the needs of the field, such as methods for *feature selection* in combination with the classification of high-dimensional data, the introduction of *string kernels* to process biological sequences, or the development of methods to *learn from several kernels simultaneously* ('composite kernels'). The interested reader can find a comprehensive introduction in (Vert, 2006).

KERNEL METHODS IN COMMUNICATIONS

There are four situations that make kernel methods good candidates for use in electromagnetics (Martínez-Ramon, 2006): 1) No close solutions exist, and the only approaches are trial and error methods. In these cases, kernel algorithms can be employed to solve the problem. 2) The application requires operating in real time, and the computation time is limited. In these cases, a kernel algorithm can be trained off-line, and used in test mode in real time. The algorithms can be embedded in any hardware device. 3) Faster convergence rates and smaller errors are required. Kernel algorithms have shown superior performance in generalization ability in many problems. Also, the block optimization and the uniqueness of solutions make kernelized versions of linear algorithms (as SVM) faster than many other methods. 4) Enough measured data exist to train a

regression algorithm for prediction and no analytical tools exist. In this case, one can actually use an SVM to solve the part of the problem where no analytical solution exist and combine the solution with other existing analytical and closed form solutions.

The use of kernelized SVMs has been already proposed to solve a variety of digital communications problems. The decision feedback equalizer (Sebald & Buclew, 2000) and the adaptive multi-user detector for Code Division Multiple Access (CDMA) signals in multipath channels (Chen et al., 2001) are addressed by means of binary SVM nonlinear classifiers. In (Rahman et al., 2004) signal equalization and detection for a MultiCarrier (MC)-CDMA system is based on an SVM linear classification algorithm. Koutsogiannis et al. (2002) introduced the use of KPCA for classification and de-noising of communication signals.

KERNEL METHODS IN SIGNAL PROCESSING

Many signal processing supervised and unsupervised schemes such as discriminant analysis, clustering, principal/independent component analysis, or mutual information extraction have been addressed using kernels (see previous chapters). Also, an interesting perspective for signal processing using SVM can be found in (Mattera, 2005), which relies on a different point of view to signal processing.

The use of time series with supervised SVM algorithms has mainly focused on two DSP problems: (1) non-linear system identification of the underlying relationship between two simultaneously recorded discrete-time processes, and (2) time series prediction (Drezet and Harrison 1998; Gretton et al., 2001; Suykens, 2001). In both of them, the conventional SVR considers lagged and buffered samples of the available signals as its input vectors.

These approaches pose several problems and opportunities. First, the statement of linear signal models in the primal problem, which will be called *SVM primal signal models*, will allow us to obtain robust estimators of the model coefficients (Rojo-Álvarez et al., 2005a) in classical DSP problems, such as ARMA modeling, the γ-filter, and spectral analysis (Rojo-Álvarez et al., 2003, Camps-Valls et al., 2004, Rojo-Álvarez et al., 2004). Second, the consideration of nonlinear SVM-DSP algorithms can be addressed from two different approaches: (1) *RKHS signal models*, which state the signal model equation in the feature space (Martínez-Ramón et al., 2005), and (2) *dual signal models*, which are based on the nonlinear regression of each single time instant with appropriate Mercer's kernels (Rojo-Álvarez et al., 2005b).

KERNEL METHODS IN SPEECH PROCESSING

An interesting and active research field is that of speech recognition and speaker verification. First, there have been many attempts to apply SVMs to improve existing speech recognition systems. Ganapathiraju (2002) uses SVMs to estimate Hidden Markov Models state likelihoods, Venkataramani et al. (2003) applied SVMs to refine the decoding search space, and in (Gales and Layton, 2004) statistical models for large vocabulary continuous speech recognition were trained using SVMs. Second, early SVM approaches by Schmidt and Gish (1996), and then by Wan and Campbell (2000), used polynomial and RBF kernels to model the distribution of cepstral input vectors. Further improvements considered mapping to *feature space* using sequence kernels (Fine et al. 2001). In the case of speaker verification, the recent works of Shriberg et al. (2005) for processing high-level stylistic or lexical features are worth mentioning.

Voice processing has been performed by using KPCA. Lima et al. (2005) used sparse KPCA for voice feature extraction and then used them for speech recognition. Mak et al. (2005) used KPCA to introduce speaker adaptation in voice recognition schemes.

KERNEL METHODS IN IMAGE PROCESSING

One of the first works proposing kernel methods in the context of image processing was (Osuna *et al.*, 1997), where a face detection system was proposed. Also, in (Papagiorgiou & Poggio, 2000) a face, pedestrian, and car detection method based on SVMs and Haar wavelets to represent images was presented.

The previous global approaches demonstrated good results for detecting objects under fixed viewing conditions. However, problems occur when the viewpoint and pose vary. Different methods have been built to

tackle these problems. For instance, the *component-based* approach (Heisele *et al.*, 2001) alleviates this face detection problem. Nevertheless, the main issues in this context are related to: the inclusion of geometric relationships between components, which were partially addressed in (Mohan *et al.*, 2001) using a two-level strategy; and automatically choose components, which was improved in (Heisele *et al.*, 2002) based on the incorporation of 3D synthetic face models database. Alternative approaches, completely automatic, have been later proposed in the literature (Ullman *et al.* 2002), and kernel direct discriminate analysis (KDDA) was used by Lu et al (2006) for face recognition.

Liu et al. (2004) used KICA to model face appearance, showing that the method is robust with respect to illumination, expression and pose variations. Zheng et al. (2006) used KKCA for facial expression recognition. Another application is *object recognition*. In the special case where the objects are human faces, it opens to *face recognition*, an extremely lively research field, with applications to video surveillance and security (see http://www.face-rec.org/). For instance, (Pontil & Verri, 1998) identified objects in the COIL database (http://www1.cs.columbia.edu/CAVE/). Vaswani et al. (2006) use KPCA for image and video classification. Texture classification using kernel independent component analysis has been, for example, used by Cheng et al. (2004), and KPCA, KCCA and SVM are compared in Horikawa (2005). Finally, it is worth mentioning the *matching kernel* (Wallraven *et al.*, 2003), which uses local image descriptors; a *modified local kernel* (Boughorbel, 2005), or the *pyramid local descriptions* (Grauman & Darrell, 2005).

Kernel methods have been used in multi-dimensional images, i.e. those acquired in (relatively high) number N of spectral bands acquired from airborne or satellite sensors. Support Vector Machines (SVMs) were first applied to hyperspectral image classification in (Gualtieri & Cromp, 1998) and their capabilities were further analyzed in (Camps-Valls et al., 2004) in terms of stability, robustness to noise, and accuracy. Some other kernel methods have been recently presented to improve classification, such as the kernel Fisher discriminant (KFD) analysis (Dundar & Langrebe, 2004), or Support Vector Clustering (SVC) (Song, Cherian, & Fan, 2005). In (Camps-Valls & Bruzzone, 2005), an extensive comparison of kernel-based classifiers was conducted in terms of the accuracy of methods when working in noisy environments, high input dimension, and limited training sets. Finally, a full family of *composite kernels* for efficient combination of spatial and spectral information in the scene has been presented in (Camps-Valls, 2006).

Classification of functional magnetic resonance images (fMRI) is a novel technique that may lead to a quantity of discovery tools in neuroscience. Classification in this domain is intended to automatically identify differences in distributed neural substrates resulting from cognitive tasks. The application of kernel methods has given reasonable results in accuracy and generalization ability. Recent work by Cox and Savoy (Cox and Savoy, 2003) demonstrated that linear discriminant analysis (LDA) and support vector machines (SVM) allow discrimination of 10 class visual activation patterns evoked by the visual presentation of various categories of objects on a trial-by-trial basis within individual subjects. LaConte et al., (2005) used a linear SVM for online pattern recognition of left and right motor activation in single subjects. Wang et al (Wang et al., 2004) applied an SVM classifier to detect brain cognitive states across multiple subjects. In (Martínez-Ramón et al., 2005a), a work has been presented that splits the activation maps into areas, applying a local (or base) classifier to each one. In (Koltchinskii et al, 2005), theoretical bounds on the performance of the method for the binary case have been presented, and in (Martínez-Ramón et al, 2006), a distributed boosting takes advantage of the fact that the distribution of the information in the brain is sparse.

FUTURE TRENDS

Kernel methods have been applied in bioinformatics, signal and speech processing, and communications, but there are many areas of science and engineering in which these techniques have not been applied, namely the emerging techniques of chemical sensing (such as olfaction), forecasting, remote sensing, and many others. Our prediction is that, provided that kernel methods are systematically showing improved results over other techniques, these methods will be applied in a growing amount of engineering areas, as long as to an increasing amount of activity in the areas surveyed in this chapter.

CONCLUSION

This chapter has revised the main applications encountered in the active field of machine learning known as kernel methods. This well-established field has emerged very useful in many application domains, mainly due to the versatility of the provided solutions, the possibility to adapt the method to the application field, the mathematical elegance and many practical properties. The interested reader can find more information on these application domains in (Camps-Valls, 2006), where a suite of applications and novel kernel developments are provided. The application and development of kernel methods to new fields and also the challenging questions answered so far ensure exciting results in the near future.

REFERENCES

Boughorbel, S. (2005). Kernels for image classification with Support Vector Machines, PhD Thesis, Université Paris 11, Orsay, July 2005.

Camps-Valls, G., & Bruzzone, L. (2005). Kernel-based methods for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing, 43*(6), 1351-1362.

Camps-Valls, G., Gómez-Chova, L., Calpe, J., Soria, E., Martín, J. D., Alonso, L., & Moreno, J. (2004). Robust support vector method for hyperspectral data classification and knowledge discovery. *IEEE Transactions on Geoscience and Remote Sensing, 42*(7), 1530-1542.

Camps-Valls, G., Gómez-Chova, L., Muñoz-Marí, J., Vila-Francés, J., & Calpe-Maravilla, J. (2006). Composite kernels for hyperspectral image classification. *IEEE Geoscience and Remote Sensing Letters,* 3(1), 93-97.

Camps-Valls, G., Martínez-Ramón, M., Rojo-Álvarez, J.L., and Soria-Olivas, E. (2004). Robust γ-filter using support vector machines. *Neurocomputing, 62*, 493-499.

Camps-Valls, G., Soria-Olivas, E., Perez-Ruixo, J. J., Perez-Cruz, F., Figueiras-Vidal, A.R., Artes-Rodriguez, A. (2002). Cyclosporine Concentration Prediction using Clustering and Support Vector Regression Methods. *IEE Electronics Letters*, (12), 568-570.

Camps-Valls, G., Rojo-Álvarez, J. L, and Martínez-Ramón, M. (2006). *Kernel methods in bioengineering, signal and image processing. Idea Group, Inc. Hershey, PA (USA). Nov. 2006*

Chen, S., Samingan, A. K., & Hanzo, L. (2001). Support vector machine multiuser receiver for DS-CDMA signals in multipath channels. *IEEE Transactions on Neural Networks*, *12*(3), 604 - 611.

Cheng, J., Liu, Q. & Lu, H. (2004). Texture classification using kernel independent component analysis. Proceedings of the 17th International Conference on Pattern Recognition, Vol. 1, p. 23-26 Aug. 2004 pp. 620- 623.

Cox, D. D. & Savoy, R. L. (2003). Functional Magnetic Resonance Imaging (fMRI) "brain reading": detecting and classifying distributed patterns of fMRI activity in human visual cortex. *Neuroimage*, 19(2), 261-70.

Drezet, P. and Harrison, R. (1998). Support vector machines for system identification. *UKACC International Conference on Control'98, 1*, 688-692, Swansea, U.K.

Dundar, M., & Langrebe, A. (2004). A cost-effective semi-supervised classifier approach with kernels. *IEEE Transactions on Geoscience and Remote Sensing, 42*(1), 264-270.

Fine S., Navratil J., & Gopinath, R. A. (2001). A Hybrid GMM/SVM Approach to Speaker Identification. Proc. IEEE International Conference on Audio Speech and Signal Processing, pp. 417-420.

Gales, M. & Layton, M. (2004). SVMs, Generative Kernels and Maximum Margin Statistical Models. Beyond HMM Workshop on Statistical Modelling Approach for Speech Recognition.

Ganapathiraju, A. (2002). Support Vector Machines for Speech Recognition. Ph.D. thesis, Mississippi State University.

Grauman, K. & Darrell, T (2005). The Pyramid Match Kernel: Discriminative Classification with Sets of Image Features. In Proceedings of the IEEE International Conference on Computer Vision, Beijing, China.

Gretton, A., Doucet, A., Herbrich, R., Rayner, P., and Schölkopf, B. (2001). Support vector regression for black-box system identification. *In 11th IEEE Workshop on Statistical Signal Processing*, 341-344, NY.

Gualtieri, J. A., & Cromp, R. F. (1998). Support Vector Machines for Hyperspectral Remote Sensing Classification, *27'th AIPR Workshop, Proceedings of the SPIE* Vol. 3584, 221- 232.

Heisele, B., Verri, A., & Poggio, T. (2002). Learning and vision machines. Proc. of the IEEE, 90(7), 1164-1177.

Horikawa, Y., (2005). Modification of correlation kernels in SVM, KPCA and KCCA in texture classification, 2005 IEEE International Joint Conference on Neural Networks. IJCNN '05. Proceedings. Vol. 4, 31 July-4 Aug. 2005, pp. 2006- 2011.

Koltchinskii, V., Martínez-Ramón, M., Posse, S., 2005. Optimal aggregation of classifiers and boosting maps in functional magnetic resonance imaging. In: Saul, L. K., Weiss, Y., Bottou, L. (Eds.), Advances in Neural Information Processing Systems 17. MIT Press, Cambridge, MA, pp. 705–712.

Koutsogiannis, G.S. & Soraghan, J., (2002) Classification and de-noising of communication signals using kernel principal component analysis (KPCA), IEEE International Conference on Acoustics, Speech, and Signal Processing, 2002. Proceedings. (ICASSP '02). Vol 2, 2002, pp. 1677-1680.

LaConte, S., Strother, S., Cherkassky, V., J. Anderson, & Hu, X. (2005). Support vector machines for temporal classification of block design fmri data. *Neuroimage*, 26, 317-329.

Lima, A., Zen, H., Nankaku, Y., Tokuda, K., Kitamura, T. & Resende, F.G. (2005) Sparse KPCA for Feature Extraction in Speech Recognition, . Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP '05) ,Vol. 1, March 18-23, 2005, pp. 353- 356.

Liu, Q, Cheng, J, Lu, H & Ma, S, (2004) Modeling face appearance with nonlinear independent component analysis, Sixth IEEE International Conference on Automatic Face and Gesture Recognition, Proceedings, 17-19, May, 2004 Page(s): 761- 766.

Lu, J., Plataniotis, K.N. & Venetsanopoulos, A.N. (2003). Face recognition using kernel direct discriminant analysis algorithms, *IEEE Transactions on Neural Networks*, *14*(1), 117- 126

Mak, B., Kwok, J.T. & Ho, S., (2005) Kernel Eigenvoice Speaker Adaptation, *IEEE Transactions on Speech and Audio Processing*, Vol. 13, No. 5 Part 2, Sept. 2005, pp. 984- 992.

Martínez-Ramón, M., Koltchinskii, V., Heileman, G., & Posse, S. (2005a). Pattern classification in functional MRI using optimally aggregated AdaBoosting. *In Organization of Human Brain Mapping, 11th Annual Meeting*, 909, Toronto, Canada.

Martínez-Ramón, M., Koltchinskii, V., Heileman, G., & Posse, S. (2005b). Pattern classification in functional mri using optimally aggregated AdaBoost. *In Proc. International Society for Magnetic Resonance in Medicine, 13th Scientific Meeting*, Miami, FL, USA.

Martinez-Ramon, M & Christodoulou, C. (2006a). Support Vector Machines for Antenna Array Processing and Electromagnetics, Morgan & Claypool, CA, USA, 2006.

Martínez-Ramón, M., Koltchinskii, V., Heileman, G., & Posse, S. (2006b). fMRI pattern classification using neuroanatomically constrained boosting. *Neuroimage*, *31*(3), 1129-1141.

Mattera, D. (2005). Support Vector Machines for Signal Processing. *In Support Vector Machines: Theory and Applications*. Lipo Wang (Ed.), Springer.

Mikolajczyk, K., & Schmid, C. (2003). A performance evaluation of local descriptors. In Proceedings of the *Conference on Computer Vision and Pattern Recognition, Madison, Wisconsin USA*, (2) 257-263.

Mohan, A., Papageorgiou, C., & Poggio, T.(2001). Example-based object detection in images by components. IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(4), 349-361.

Mukherjee, S., Tamayo, P., Mesirov, J. P., Slonim, D., Verri, A., and Poggio, T. (1998). *Support vector machine classification of microarray data.* Technical Report 182, C.B.L.C. A.I. Memo 1677.

Osowski, S., Hoai, L. T., & Markiewicz, T. (2004). Support vector machine-based expert system for reliable heartbeat recognition. *IEEE Trans Biomed Eng, 51*(4), 582-9.

Osuna, E., Freund, R., & Girosi, F. (1997). Training Support Vector Machines: an application to face detection. In P*roceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Puerto Rico, 17-19.

Papageorgiou, C. & Poggio, T. (2000) A trainable system for object detection. International Journal of Computer Vision, 38(1), pp 15-33.

Pontil, M. & Verri A. (1998) Support Vector Machines for 3D object recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 6, 637-646.

Rahman, S., Saito, M., Okada, M., & Yamamoto, H. (2004). An MC-CDMA signal equalization and detection scheme based on support vector machines. *Proc. of 1st Int Symp on Wireless Comm Syst*, *1*, 11 - 15.

Rojo-Álvarez, J. L., Camps-Valls, G., Martínez-Ramón, M., Soria-Olivas, E., A. Navia Vázquez, & Figueiras-Vidal, A. R. (2005). Support vector machines framework for linear signal processing. *Signal Proccessing*, *85*(12), 2316 – 2326.

Rojo-Álvarez, J. L., Martínez-Ramón, M., Figueiras-Vidal, A. R., García-Armada, A., and Artés-Rodríguez, A. (2003). A robust support vector algorithm for nonparametric spectral analysis. *IEEE Signal Processing Letters, 10*(11), 320-323.

Rojo-Álvarez, J., Figuera, C., Martínez-Cruz, C., Camps-Valls, G., and Martínez-Ramón, M. (2005). Sinc kernel nonuniform interpolation of time series with support vector machines. Submitted.

Rojo-Álvarez, J., Martínez-Ramón, M., Figueiras-Vidal, A., dePrado Cumplido, M., and Artés-Rodríguez, A. (2004). Support vector method for ARMA system identification. *IEEE Transactions on Signal Procesing 52*(1), 155-164.

Schmidt, M., & Gish, H. (1996). Speaker Identification via Support Vector Classifiers. Proc. IEEE International Conference on Audio Speech and Signal Processing, pp. 105-108.

Schölkopf, B., & Smola, A. (2002). *Learning with kernels.* Cambridge, MA: MIT Press.

Sebald, D. & Buclew, A. (2000). Support vector machine techniques for nonlinear equalization. *IEEE Transactions on Signal Processing*, *48*(11), 3217 - 3226.

Shriberg, E., Ferrer, L., Kajarekar, S., Venkataraman, A., & Stolcke, A. (2005). Modelling Prosodic Feature Sequences for Speaker Recognition. *Speech Communication* 46, pp. 455-472. Special Issue on Quantitative Prosody Modelling for Natural Speech Description and Generation.

Song, X., Cherian, G., & Fan, G. (2005). A ν-insensitive SVM approach for compliance monitoring of the conservation reserve program. *IEEE Geoscience and Remote Sensing Letters, 2*(2), 99-103.

Srivastava, A. N., & Stroeve, J. (2003). Onboard detection of snow, ice, clouds and other geophysical processes using kernel methods. In *Proceedings of the ICML 2003 Workshop on Machine Learning Technologies for Autonomous Space Sciences*. Washington, DC USA.

Ullman, S., Vidal-Naquet, M., & Sali, E. (2002) Visual features of intermediate complexity and their use in classification. Nature Neuroscience, 5(7), 1-6.

Vaswani, N. & Chellappa, R. (2006) Principal components space analysis for image and video classification, IEEE Transactions on Image Processing, Vol. 15 No 7, July 2006, pp.1816- 1830.

Venkataramani, V., Chakrabartty, S., & Byrne, W. (2003). Support Vector Machines for Segmental Minimum Bayes Risk Decoding of Continuous Speech. In Proc. IEEE Automatic Speech Recognition and Understanding Workshop.

Vert, Jean-Philippe (2006). *Kernel methods in genomics and computational biology.* In book: "Kernel methods in bioengineering, signal and image processing". Eds.: G. Camps-Valls, J. L Rojo-Álvarez, M. Martínez-Ramón. Idea Group, Inc. Hershey, PA. USA.

Wallraven, C., Caputo, B., & Graf, A. (2003) Recognition with local features: the kernel recipe. In Proceedings of the IEEE International Conference on Computer Vision, Nice, France.

Wan, V., & Campbell, W. M. (2000). Support Vector Machines for Speaker Verification and Identification. Proc. Neural Networks for Signal Processing X, pp. 775-784.

Wang, X., Hutchinson, R., & Mitchell, T. M. (2004). Training fMRI classifiers to discriminate cognitive states across multiple subjects. *In Thrun, S., Saul, L., and Schölkopf, B., editors, Advances in Neural Information Processing Systems* 16. MIT Press, Cambridge, MA.

Zheng, W., Zhou, X., Zou, C., & Zhao, L., (2006) Facial expression recognition using kernel canonical correlation analysis (KCCA) IEEE Transactions on Neural Networks, Vol. 17, No 1, Jan. 2006, pp. 233- 238.

KEY TERMS

Bioinformatics: This is the application of informatics to analysis of experimental data and simulation of biological systems.

Biomedicine: This refers to the application of engineering to the medicine. It involves the design of medical equipment, prosthesis, and systems and algorithms for diagnose and therapy.

Communications: Communication is the act of sending a message to one or more receivers. In this context, we use the word communication to refer to the technologies and theory of telecommunications engineering.

Composite Kernels: A composite kernel (see chapter 1 for a definition of kernel) is a linear combination of several Mercer kernels $K_i(\mathbf{x}_i,\mathbf{x}_j)$ of the form

$$K\left(\mathbf{x}_i,\mathbf{x}_j\right)=\sum_{i=1}^{L} a_i K_i\left(\mathbf{x}_i,\mathbf{x}_j\right)$$

For the composite kernel to be a Mercer kernel, it needs to be semi definite positive. A sufficient condition for a linear combination of Mercer kernels to be a valid Mercer kernel is simply $a_i \geq 0$.

Image Processing: This term refers to any manipulation of digital images in order to compress/decompress, transfer it through a communications channel or alter its properties, such as color, textures, definition, etc. More interestingly, image processing includes all techniques used in order to obtain information embedded in a digital image or a sequence of digital images, for example, detection and classification of objects or events into a sequence of images.

Kernel Methods: A shortened name for Kernel-based learning methods (see previous chapter for an introduction to kernel methods). Kernel methods include all machine learning algorithms that are intrinsically linear but, through a nonlinear transformation of the input data into a highly dimensional Hilbert space, present nonlinear properties from the point of view of the input data. The Hilbert space must be provided with an inner product that can be expressed as a function of the input data itself, thus avoiding the explicit use of vectors in these spaces. Such an inner product satisfies the so called Mercer conditions and is called a Mercer Kernel.

Signal Processing: It is the analysis, interpretation and manipulation of signals. Usually, one calls signal to a set of data that has been collected sequentially and thus it has temporal properties. Signal processing includes, among others, storage, reconstruction, detection of information in presence of noise, interferences or distortion, compression, encoding, decoding and many other processing techniques that may involve machine learning.

Speech Processing: Signal processing of speech signals. An important group of speech processing techniques are those devoted to speech communications. Here, the processing includes analog to digital and digital to analog conversions, and compression/decompression algorithms that usually use speech modeling through autoregressive models of small voice frames, exploiting their local stationarity properties. Another important block of techniques is the speech recognition, which involves machine learning techniques. Traditionally, Hidden Markov Models have been used for speech recognition, but recently kernel methods have been used with promising results. Also, there is an intense research in text-to-speech conversion.

Support Vector Machines (SVM): A SVM is a linear learning machine constructed through an algorithm that uses an optimization criterion which is based on the compromise between the training error and the complexity of the resulting learning machine. The optimization criterion for classification is

$$L=\frac{1}{2}\|\mathbf{w}\|^2+C\sum_{i=1}^{N}\xi_i$$

subject to

$$y_i\left(\mathbf{w}^T\mathbf{x}_i+b\right)>1-\xi_i$$
$$\xi_i \geq 0$$

where $\mathbf{w}$ are the parameters of the learning machine, $\mathbf{x}_i$ are the training data and y_i are their corresponding labels, and ξ_i are the so called slack variables. The criterion minimizes the training data classification error plus the complexity of the machine through the minimization of the norm of the machine parameters. This is equivalent to maximize the generalization properties of the machine. The resulting parameters are expressed as a linear combination of a subset of the training data (the support vectors). This algorithm is easily extensible to a nonlinear algorithm using the kernel trick, reason for what it is usually considered a kernel method. A similar algorithm is used for regression.

Architecture for Symbolic Object Warehouse

Sandra Elizabeth González Císaro
Universidad Nacional del Centro de la Provincia de Buenos Aires, Argentina

Héctor Oscar Nigro
Universidad Nacional del Centro de la Provincia de Buenos Aires, Argentina

INTRODUCTION

Much information stored in current databases is not always present at necessary different levels of detail or granularity for Decision-Making Processes (DMP). Some organizations have implemented the use of central database - Data Warehouse (DW) - where information performs analysis tasks. This fact depends on the Information Systems (IS) maturity, the type of informational requirements or necessities the organizational structure and business own characteristic.

A further important point is the *intrinsic structure of complex data*; nowadays it is very common to work with complex data, due to syntactic or semantic aspects and the processing type (Darmont et al., 2006). Therefore, we must design systems, which can to maintain data complexity to improve the DMP.

OLAP systems solve the problem of present different aggregation levels and visualization for multidimensional data through cube's paradigm. The classical data analysis techniques (factorial analysis, regression, dispersion, etc.) are applied to individuals (tuples or individuals in transactional databases). The classic analysis objects are not expressive enough to represent tuples, which contain distributions, logic rules, multivaluate attributes, and intervals. Also, they must be able to respect their internal variation and taxonomy maintaining the dualism between individual and class.

Consequently, we need a new data type holding these characteristics. This is just the mathematical concept model introduced by Diday called Symbolic Object (SO). SO allows modeling physic entities or real world concepts. The former are the tuples stored in transactional databases and the latter are high entities obtained from expert's analysis, automatic classification or some particular aggregation taken from analysis units (Bock & Diday, 2000).

The SO concept helps construct the DW and it is an important development for Data Mining (DM): for the manipulation and analysis of aggregated information (Nigro & González Císaro, 2005). According to Calvanese, data integration is a central problem in the design of DWs and Decision Support Systems (Calvanese, 2003; Cali, et al., 2003); we make the architecture for Symbolic Object Warehouse construction with integrative goal. Also, it combines with Data Analysis tasks or DM.

This paper is presented as follows: First, Background: DW concepts are introduced. Second, Main Focus divided into: SOs Basic Concepts, Construing SOs and Architecture. Third, Future Trends, Conclusions, References and Key Terms.

Background

The classical definition given by the theme's pioneer is "a Data Warehouse is a subject-oriented, integrated, time-variant, and non-volatile collection of data in support of management's Decision-Making Process" (Inmon, 1996). The fundamental purpose of a DW is to empower the business staff with information that allows making decisions based on consolidated information. In essence, a DW is in a continuous process of transformation as regards information and business rules; both of them must be considered at design time to assure increase robustness and flexibility of the system.

Extraction, Transformation and Load (ETL) constitute the fundamental process in the DW. It is liable for the extraction of data from several sources, their cleansing, customization and insertion into a DW (Simitsis, et al., 2005). When complex data is involved, this process becomes difficult, because of the integration of different semantics (especially with text data, sound, images, etc) or complex structures. So, it is necessary to include integration functions able to join and to merge them.

Metadata management, in DW construction, helps the user understand the stored contents. Information about the meaning of data elements and the availability of reports are indispensable to successfully use the DW.

The generation and management of metadata serve two purposes (Staudt et al., 1999):

1. To minimize the efforts for development and administration of a DW
2. To improve the extraction from it.

Web Warehouse (WW) is a major topic widely researched and developed (Han & Kamber, 2001), as a result of the increasing and intensive use in e-commerce and e-business applications. WW tools and applications are morphing into enterprise portals and analytical applications are being extended to transactional systems. With the same direction, the audiences for WW have expanded as analytical applications have rapidly moved (indirectly) into the transactional world ERP, SCM and CRM (King, 2000).

Spatial data warehousing (SDW) responds to the need of providing users with a set of operations for easily exploring large amounts of spatial data, as well as for aggregating spatial data into synthetic information most suitable for decision-making (Damiani & Spaccapietra, 2006). Gorawski & Malczok (2004) present a distributed SDW system designed for storing and analyzing a wide range of spatial data. The SDW works with the new data model called cascaded star model that allows efficient storing and analyzes of huge amounts of spatial data.

MAIN FOCUS

SOs Basic Concepts

Formally, a SO is a triple s = (a, R, d) where R is a relation between descriptions, d is a description and "a" is a mapping defined from Ω(discourse universe) in L depending on R and d (Diday, 2003).

According to Gowda's definition: "SOs are extensions of classical data types and they are defined by a logical conjunction of events linking values and variables in which the variables can take one or more values, and all the SOs need not be defined on the same variables" (Gowda, 2004). We consider SOs as a new data type for complex data define algebra at Symbolic Data Analysis.

An SO models an individual or a class maintaining its taxonomy and internal variation. In fact, we can represent a concept by its intentional description, i.e. the necessary attributes to characterize to the studied phenomenon and the description allows distinguishing ones from others.

The key characteristics enumerated by Gowda (2004) that do SO a complex data are:

- All objects of a symbolic data set may not be defined on the same variables.
- Each variable may take more than one value or even an interval of values.
- In complex SOs, the values, which the variables take, may include one or more elementary objects.
- The description of an SO may depend on the existing relations between other objects.
- The descriptor values may have typicality values, which indicate frequency of occurrence, relative likelihood, level of importance of the values, …

There are two main kinds of SOs (Diday & Billard, 2002):

- *Boolean SOs*: The instance of one binary relation between the descriptor of the object and the definition domain, which is defined to have values true or false. If [y (w) R d] = {true, false} is a Boolean SO. Example: s=(pay-mode ∈ {good; regular}), here we are describing an individual/class of customer whose payment mode is good or regular.
- *Modal SOs*: In some situations, we cannot say true or false, we have a degree of belonging, or some linguistic imprecision as always true, often true, fifty-fifty, often false, always false; here we say that the relation is fuzzy. If [y (w) R d] ∈ L = [0,1] is a Modal SO. Example: s=(pay-mode ∈ [(0.25) good; (0.75) regular]), at this point we are describing an individual/class of customer that has payment mode: 0.25 good; 0.75 regular.

The *SO extension* is a function that helps recognize when an individual belongs to the class description or a class fits into a more generic one. In the Boolean case,

the extent of an SO is denoted Ext(s) and defined by the extent of "a", which is: Extent (a) = {w ∈ Ω / a (w) = true}. In the Modal instance, given a threshold α, it is defined by Ext_{α} (s)= $Extent_{\alpha}$ (a)= {w ∈ Ω / a (w) ≥ α}.

It is possible to work with SOs in two ways:

- Induction: We know values of their attributes then we know what class they belong.
- Generalization: We want to form a class from the generalization/specialization process of the values of the attributes of a set of individuals.

There is an important number of methods (Bock et al, 2000) developed to analyze SO, which were implemented in Sodas 1.2 and Sodas 2.5 software through Sodas and Asso projects respectively; whose aim is to analyze official data from Official Statistical Institutions (see ASSO or SODAS Home Page).

The principal advantages in SO use are (Bock & Diday, 2000; Diday, 2003):

- It preserves the confidentiality of the information.
- It supports the initial language in the one that SOs were created.
- It allows the spread of concepts between Databases.
- Being independent from the initial table, they are capable of identifying some individual coincidence described in another table.

As a result of working with higher units called *concepts* necessarily described by more complex data, DM is extended to Knowledge Mining (Diday 2004).

Construing SOs

Now we are going to create SOs, Let's suppose we want to know client's profile grouped by work's activity. How do we model this kind of situations with SOs?

The SOs descriptor must have the following attributes:

1. Continent
2. Age
3. Study Level

Suppose that in our operational databases we have stored the relational Tables 1 and 2.

Notice we take an SO as every value of the variable work activity. The SOs descriptors are written in the same notation used in Bock and Diday's book:

SO-Agriculture(4)=[Study Level={"low"(0.50), "medium"(0.50)}] ∧ [Continent = {"America"(0.5), "Europe"(0.25), "Oceania"(0.25)}] ∧ [Age= [30:42]]].
SO-Manufactures (3) = [Study Level ={"low"(0.33), "medium"(0.33), "high"(0.33)}] ∧ [Continent = {"Asia"(0.33), "Europe"(0.66)}] ∧ [Age= [28:50]]].

Table 1. Customer

#Customer	…	Initial Transaction	Age	Country	Study Level	Work's Activity
041	…	23-May-03	50	Spain	Medium	Manufactures
033	…	25-Jul-03	45	China	High	Manufactures
168	…	30-Jul-03	30	Australia	Low	Agriculture
457	…	2-Jan-04	39	Sudan	High	Services
542	…	12-Feb-04	35	Argentina	Medium	Agriculture
698	…	13-April-04	48	India	High	Services
721	…	22-Aug-04	60	France	High	Services
844	…	15-Sep-04	53	Canada	Medium	Services
987	…	25-Oct-04	42	Italy	Low	Agriculture
1002	…	10-Nov-04	28	Germany	Low	Manufactures
1299	…	28-Dec-04	34	EEUU	Medium	Agriculture

Table 2. Taxonomy

Country	Continent
Spain	Europe
China	Asia
Australia	Oceania
Sudan	Africa
Argentina	America
India	Asia
France	Europe
Canada	America
Italy	Europe
Germany	Europe
EEUU	America

SO-Services (4) [Study Level = {"medium"(0.25), "high"(0.75)}] ∧ [Continent = {"Africa" (0.25), "America"(0.25), "Asia"(0.25), "Europe"(0.25)}] ∧ [Age= [39:60]]].

Now we have second order units representing the concept activity of our clients. The number in brackets is the quantity of individuals belonging to the SO, the variables show the values for the class, for example SO-Manufactures: the variable Study Level shows equal probability. The clients are distributed 33 % in Asia and 66 % in Europe. The age is between 39 and 60 years.

To plan the analysis units or SOs we need:

- Knowledge domain,
- Rules of the business,
- Type of information stored in the operational systems, -organizational structures.

We call the former elements *Background Knowledge*.

Architecture

Figure 1 shows the information flows, information knowledge and the most important tasks covered by this architecture (González Císaro, Nigro & Xodo, 2006). Generally, almost all current DW and DM solutions are based on decoupled architectures. DM tools suppose the data to be already selected, cleaned and transformed. Solutions integrating steps must be addressed.

Figure 2 shows a conceptual architecture to identify the most important modules of the system. A manager is associated to each of them, so that they achieve flexibility (it is simple to add new functions); and the functionality encapsulation in every component helps the design organization and modularization. Thus, we can distinguish:

- System functionalities.
- What component carries out each task
- Information/knowledge workflows.

In the next paragraphs, a briefly explanation of each component functionality is completed.

Intelligent Interface: It is responsible for the connection between the system and the user. We design this component with two *Intelligent Discovery Assistants* (Bernstein et al., 2005); one assists in DW tasks and the other with analysis or DM.

ETL Manager: The user defines the SO descriptor and the system must obtain the data from operational databases and external sources. Two different types loads are assumed:

- Initial a predefined SO descriptor, which models the principal business concepts.
- *Ad hoc* with new SOs, which respond to new informational requirements.

The major sub components of ETL Manager module are:

- **ETL Scheduler**
- **Extraction Engine &Load Engine**
- **Transformation & Clean Engine**

Mining & Layout Manager: It is the core analysis. It shows SOs descriptors and makes all type of graphics. Particularly, graphic subcomponent has to implement Zoom Star graphic (Noirhomme, 2000, 2004), which is the best way to visualize SOs. The main subcomponents are:

- **Mining Scheduler**
- **Method Engine**
- **Method DB**
- **Graphic Manager**
- **Exploration Manager**

Figure 1. Information & knowledge flow

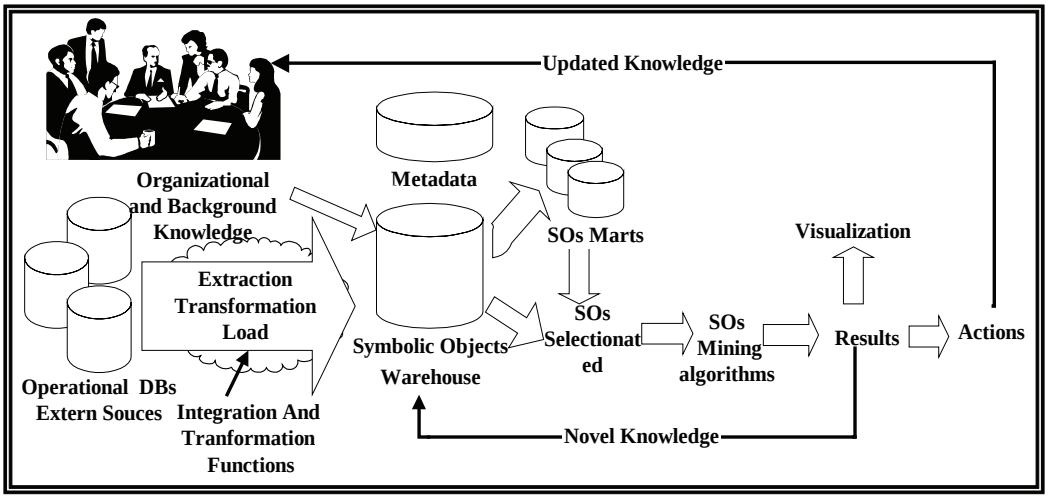
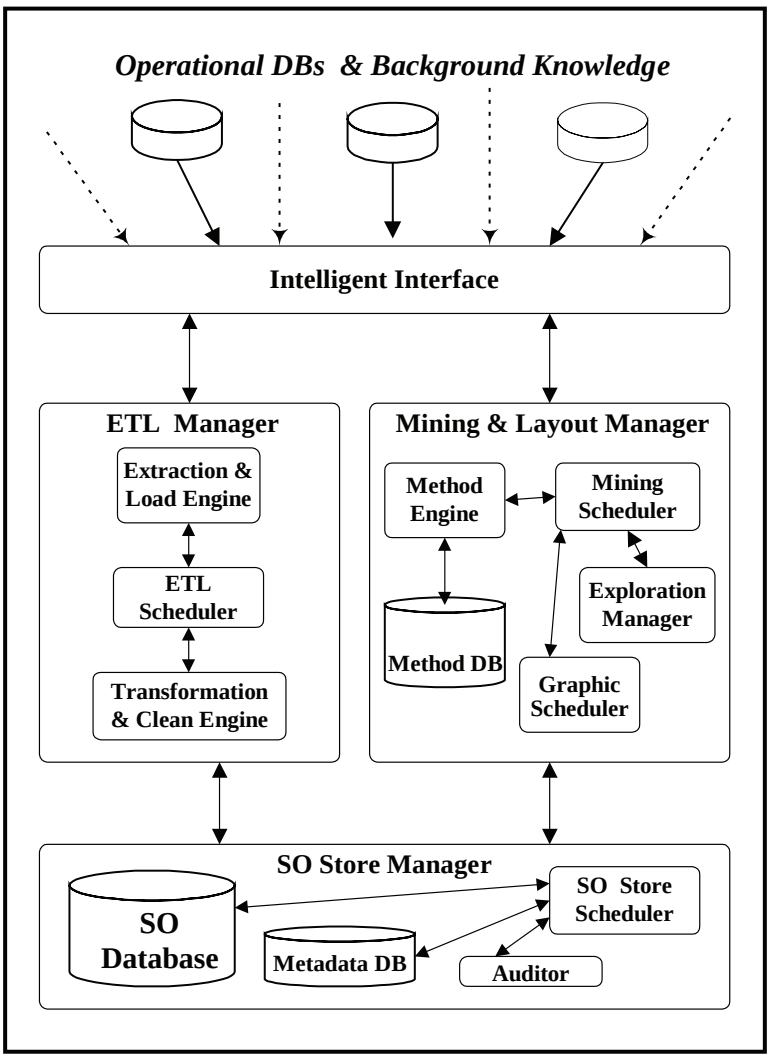


Figure 2. Conceptual architecture



SO Store Manager: It stores the SOs, SOs metadata, does concurrence control, audits and it is safe. Also, the component logs, controls, assigns and changes roles with the users.

Metadata for SOs, as Vardaki (2004) affirms, should describe the symbolic variables, their nature, components and domain. All metadata necessary for Symbolic data creation and processing can be presented as a metadata template or modeled in a separate metadata schema. The advance of the modeling process is that it will indicate not only the metadata items considered and in a structured format, specify their relation and the operators/transformations that can be applied for further manipulations. In this architecture, an independently adopted schema to store metadata about SOs- Metadata DB was adopted.

The SO Store Management has four key subcomponents:

- ***SO & Metadata Scheduler***
- ***SO Database***
- ***Metadata DB***
- ***Auditor***

Future Trends

The next step is the formal specification of the architecture in terms of design. The problems to be resolved are:

- The construction of a language to manipulate SOs.
- How to store SOs since temporary and spatial efficiency is necessary.

Given the functional modularity, an object-oriented implementation would be the most suitable. Another implementation that would be very attractive is through a multi-agents system.

Potential progress in the algorithms that work on SOs will be guided by the techniques to be explored and developed. The most important and useful in DM are: Association Rules, Regressions, Cluster Interpretability and other types of Neuronal Networks.

CONCLUSION

An SO allows representing physics entities or real word concepts in dual form, respecting their internal variations and structure. The SO Warehouse permits the intentional description of most important concepts by means of the initial language users make use of.

The quality control, security and accuracy of information are obtained in SO creation processes, since the null values means are established in this process and the metadata are included (the latter are especially important in DW and the DMP).

One of the most valued advantages in the use of SO is the capacity to carry out various levels of analysis, with which the output of one method is the input of the other. This can be observed in clustering or classification methods, as in most cases the output is a SOs set.

The principal disadvantages arisen by the use of SOs are:

- The complexity in the determination of whom will be the best SOs that will represent the analysis tasks in the organization.
- When to update or to change SOs.

As a result of the flexibility and modularity of its design, our architecture allows an integrated environment of work, with possibilities of improvement and growth. As regards Symbolic Data Analysis, DW & DM integration is very important since it can be very practical to add the discovered knowledge into DW. We discover new potential clients characteristics or relations thus SO descriptors in DW can be updated, creating new SOs. Therefore, the work with higher units like SOs could improve Knowledge Management and Decision-Making Processes.

REFERENCES

ASSO, Project Home Page. Retrieved May 2006, from http://www.info.fundp.ac.be/asso/.

Bernstein, A., Provost, F. & Hill, S. (2005).”Towards Intelligent Assistance for a Data Mining Process: An Ontology-based Approach for Cost-sensitive Classification”, *IEEE Transactions on Knowledge and Data Engineering,* 17(4), pp 503-518.

Bock, H. & Diday, E. (2000) *Analysis of Symbolic Data. Studies in Classification, Data Analysis and Knowledge Organization*. Heidelberg, Germany. Springer Verlag-Berlin.

Cali, A., Lembo, D., Lenzerini, M. & Rosati, R. (2003). Source Integration for Data Warehousing. In Rafanelli M. (Ed.), *Multidimensional Databases: Problems and Solutions* (pp. 361-392), Hershey, PA: Idea Group Publishing

Calvanese, D. (2003) Data integration in Data Warehousing. Invited talk presented at *Decision Systems Engineering Workshop (DSE'03),* Velden, Austria.

Damiáni, M. & Spaccapietra, S. (2006) Spatial Data in Warehouse Modeling. In Darmont, J. & Boussaid, O. (Eds) *Processing and Managing Complex Data for Decision Support* (pp. 1-27). Hershey, PA: Idea Group Publishing.

Darmont, J. & Boussaid, O. (2006). *Processing and Managing Complex Data for Decision* Support. Hershey, PA: Idea Group Publishing.

Diday, E. & Billard, L. (2002). *Symbolic Data Analysis: Definitions and examples*. Retrieved March 27, 2006, from http://www.stat.uga.edu/faculty/LYNNE/tr_symbolic.pdf.

Diday, E. (2003). Concepts and Galois Lattices in Symbolic Data Analysis. *Journées de l'Informatique Messine. JIM'2003. Knowledge Discovery and Discrete Mathematics* Metz, France.

Diday, E. (2004). From Data Mining to Knowledge Mining: Symbolic Data Analysis and the Sodas Software. *Proceedings of the Workshop on Applications of Symbolic Data Analysis.* Lisboa Portugal. Retrieved January 25, 2006, from http://www.info.fundp.ac.be/asso/dissem/W-ASSO-Lisbon-Intro.pdf

González Císaro, S., Nigro, H. & Xodo, D.(2006, February). Arquitectura conceptual para Enriquecer la Gestión del Conocimiento basada en Objetos Simbólicos. In Feregrino Uribe, C., Cruz Enríquez, J. & Díaz Méndez, A. (Eds.) *Proceeding of V Ibero-American Symposium on Software Engineering* (pp. 279-286), Puebla, Mexico.

Gowda, K. (2004). Symbolic Objects and Symbolic Classification. Invited paper in *Proceeding of Workshop on Symbolic and Spatial Data Analysis: Mining Complex Data Structures*. ECML/PKDD. Pisa, Italy.

Gorawski, M., Malczok, R. (2003). Distributed Spatial Data Warehouse. 5th International Conference on Parallel Processing and Applied Mathematics, Cz_stochowa, Springer Verlag, LNCS3019.

Han, J. & Kamber, M. (2001). Data Mining: Concepts and Techniques, San Francisco: Morgan Kaufmann.

Inmon, W. (1996). Building the Data Warehouse (2nd edition). New York: John Wiley & Sons, Inc.

King, D. (2000). Web Warehousing: Business as Usual? In DM Review Magazine. May 2000 Issue.

Nigro, H. & González Císaro, S. (2005). Symbolic Object and Symbolic Data Analysis. In Rivero, L., Doorn, J. & Ferraggine, V. (Eds.) *Encyclopedia of Database Technologies and Applications*. Hershey, PA: Idea Group Publishing, p. 665-670.

Noirhomme, M. (2004, January). Visualization of Symbolic Data. Paper presented at *Workshop on Applications of Symbolic Data Analysis*. Lisboa Portugal.

Sodas Home Page. Retrieved August 2006, from http://www.ceremade.dauphine.fr/~touati/sodas-pagegarde.htm.

Simitsis, A., Vassiliadis, P., Sellis, T., (2005). Optimizing ETL Processes in Data Warehouses. In *Proceedings of the 21st IEEE International Conference on Data Engin*eering (pp. 564-575), Tokyo, Japan.

Staudt, M., Vaduva, A. and Vetterli, T. (1999). *The Role of Metadata for Data Warehousing*. Technical Report of Department of Informatics (IFI) at the University of Zurich, Swiss.

Vardaki, M. (2004). Metadata for Symbolic Objects. *JSDA Electronic Journal of Symbolic Dat*a Analysis-2(1). ISSN 1723-5081.

KEY TERMS

Cascaded Star Model: A main fact table. The main dimensions form smaller star schemas in which some dimension tables may become a fact table for other, nested star schemas.

Customer Relationship Management (CRM): A methodology used to learn more about customers' wishes and behaviors in order to develop stronger relationships with them.

Enterprise Recourse Planning (ERP): A software application that integrates planning, manufacturing, distribution, shipping, and accounting functions into a single system, designed to serve the needs of each different department within the enterprise.

Intelligent Discovery Assistant: Helps data miners with the exploration of the space of valid DM processes. It takes advantage of an explicit ontology of data-mining techniques, which defines the various techniques and their properties. (Bernstein et al, 2005, pp 503-504).

Knowledge Management: An integrated, systematic approach to identifying, codifying, transferring, managing, and sharing all knowledge of an organization.

Supply Chain Management (SCM): The practice of coordinating the flow of goods, services, information and finances as they move from raw materials to parts supplier to manufacturer to wholesaler to retailer to consumer.

Symbolic Data Analysis: A relatively new field that provides a range of methods for analyzing complex datasets. It generalizes classical methods of exploratory, statistical and graphical data analysis to the case of complex data. Symbolic data methods allow the user to build models of the data and make predictions about future events (Diday 2002).

Zoom Star: A graphical representation for SO where each axis correspond to a variable in a radial graph. Thus it allows variables with intervals, multivaluate values, weighted values, logical dependences and taxonomies to be represented. A 2D and 3D representation have been designed allowing different types of analysis.

Association Bundle Identification

Wenxue Huang
Generation5 Mathematical Technologies, Inc., Canada

Milorad Krneta
Generation5 Mathematical Technologies, Inc., Canada

Limin Lin
Generation5 Mathematical Technologies, Inc., Canada
Mathematics and Statistics Department, York University, Toronto, Canada

Jianhong Wu
Mathematics and Statistics Department, York University, Toronto, Canada

INTRODUCTION

An association pattern describes how a group of items (for example, retail products) are statistically associated together, and a meaningful association pattern identifies 'interesting' knowledge from data. A well-established association pattern is the *association rule* (Agrawal, Imielinski & Swami, 1993), which describes how *two sets of items* are associated with each other. For example, an association rule *A-->B* tells that 'if customers buy the set of product *A*, they would also buy the set of product *B* with probability greater than or equal to *c*'.

Association rules have been widely accepted for their simplicity and comprehensibility in problem statement, and subsequent modifications have also been made in order to produce more interesting knowledge, see (Brin, Motani, Ullman and Tsur, 1997; Aggarwal and Yu, 1998; Liu, Hsu and Ma, 1999; Bruzzese and Davino, 2001; Barber and Hamilton, 2003; Scheffer, 2005; Li, 2006). A relevant concept is the *rule interest* and excellent discussion can be found in (Shapiro 1991; Tan, Kumar and Srivastava, 2004). Huang et al. recently developed *association bundles* as a new pattern for association analysis (Huang, Krneta, Lin and Wu, 2006). Rather than replacing the association rule, the association bundle provides a distinctive pattern that can present meaningful knowledge not explored by association rules or any of its modifications.

BACKGROUND

Association bundles are important to the field of Association Discovery. The following comparison between association bundles and association rules support this argument. This comparison is made with focus on the association structure.

An *association structure* describes the structural features of an association pattern. It tells how many association relationships are presented by the pattern, and whether these relationships are asymmetric or symmetric, between-set or between-item. For example, an association rule contains one association relationship, and this relationship exists between two sets of item, and it is asymmetric from the rule antecedent to the rule consequent. However, the asymmetric between-set association structure limits the application of association rules in two ways. Firstly, when reasoning based on an association rule, the items in the rule antecedent (or consequent) must be treated as whole -a combined item, not as individual items. One can not reason based on an association rule that a certain individual antecedent item, as one of the many items in rule antecedent, is associated with any or all of the consequent items. Secondly, one must be careful that this association between the rule antecedent and the rule consequent is asymmetric. If the occurrence of the entire set of antecedent items is not deterministically given, for example, the only given information is that a customer has chosen the consequent items, not the antecedent items, it is highly probably that she/he does not chose any of the antecedent items. Therefore, for applications where between-item

symmetric associations are required, for example, cross selling a group of items by discounting on one of them, association rules cannot be applied.

The association bundle is developed to resolve the above problems by considering the symmetric pair-wise between-item association structure. There are multiple association relationships existing in an association bundle - every two bundle-elements are associated with each other, and this between-element association is symmetric – there is no difference between the two associated items in terms of antecedence or consequence. With the symmetric between-element association structure, association bundles can be applied to applications where the asymmetric between-set association rules fail. Association bundles support marketing efforts where the sales improvement is expected on *every element* in a product group. One such example is the shelf management. An association bundle suggests that whenever and whichever an item *i* in the bundle is chosen by customers, every other item *j* in the bundle should possibly be chosen as well, thus items from the same bundle should be put together in the same shelf. Another example is the cross-selling by discounting. Every weekend retailers print on their flyers the discount list, and if two items have strong positive correlation, they should perhaps not be discounted simultaneously. With this reasoning, an association bundle can be used to do list checking, such that only one item in an association bundle will be discounted.

PRINCIPAL IDEAS

Let S be a transaction data set of N records, and I the set of items defining S. The *probability of an item k* is defined as $Pr(k) = |S(k)| / N$, where $|S(k)|$ is the number of records containing the item k. The *joint probability of two items j and k* is defined as $Pr(j,k) = |S(j,k)| / N$, where $|S(j,k)|$ is the number of records containing both j and k. The *conditional probability of the item j with respect to the item k* is defined as $Pr(j|k) = Pr(j) / Pr(j,k)$, and the *lift the item j and k* is defined as $Lift(j,k) = Pr(j,k) / (Pr(j)*Pr(k))$.

Definition. An association bundle is a group of items $b=\{i_1,\ldots,i_m\}$, a subset of I, that any two elements i_j and i_k of b are associated by satisfying that

(i). the lift for i_j and i_k is greater than or equal to a given threshold L, that is,

$Pr(i_j, i_k) / (Pr(i_j) * Pr(i_k)) >= L$;

(ii). both conditional probabilities between i_j and i_k are greater than or equal to a given threshold T, that is,

$$Pr(i_j | i_k) >= T \quad \text{and} \quad Pr(i_k | i_j) >= T.$$

An example is shown in the Figure 1 on association bundles. Figure 1 contains six tables. The first table shows the transaction data set, which is the one that used by Agrawal etc. (Agrawal, et. al., 1994) to illustrate the identification of association rules. The second and the third tables display the between-item conditional probability and lift values, respectively. The forth table displays the item pairs that have the conditional probability and lift values greater than or equal to the given thresholds, these item pairs are associated item pairs by definition. The fifth table shows the identified association bundles. For comparison, we display the association rules in the sixth table. A comparison between the association bundles and the association rules reveals that the item set {2,3,5} is identified as an association rule but not an association bundle. Check the fourth table we can see the item pair {2,3} and the item pair {3,5} actually have the lift values smaller than 1, which implies that they are having negative association with each other.

We further introduce association bundles in the following four aspects—association measure, threshold setting of measure, supporting algorithm, and main properties—via comparisons between association bundles and association rules.

Association Measure

The conditional probability (confidence) is used as the association measure for association rules (Agrawal, Imielinski & Swami, 1993), and later other measures are introduced (Liu, Hsu and Ma, 1999, Omiecinski 2003). Detailed discussions about association measures can be found in (Tan, Kumar and Srivastava, 2004). Association bundles use the between-item lift and the between-item conditional probabilities as the association measures (Huang, Krneta, Lin and Wu, 2006). The between-item lift guarantees that there is strong positive correlation between items: the between-item conditional probabilities ensure that the prediction

of one item with respect to another is accurate enough to be significant.

Threshold Setting of Measure

The value range of the confidence threshold is defined in [0,1] in association rule mining, which is the value range of the conditional probability. In association bundles, the value ranges for threshold are determined by data. More specifically, the between-item lift threshold *L(beta)* and the between-item conditional probability threshold *T(alpha)* are defined, respectively, as,

$$L(beta)=L_A+ beta * (L_M-L_A),\ beta \text{ is in } [0,1],$$
$$T(alpha)=P_A+ alpha * (P_M-P_A),\ alpha \text{ is in } [0,1],$$

where L_A and L_M are the mean and maximum between-item lifts of all item pairs whose between-item lifts are greater than or equal to 1, P_A and P_M are the mean and maximum between-item conditional probabilities of all item pairs; and beta and alpha are defined as the Strength Level Threshold.

Supporting Algorithm

Identifying "frequent itemsets" is the major step of association rule mining, and quite a few excellent algorithms such as the ECLAT (Zaki, 2000), FP-growth (Han, Pei, Yin & Mao, 2004), Charm (Zaki & Hsiao 2002), Closet (Pei, Han & Mao, 2000) have been proposed. The identification of association bundles does not compute ``frequent itemsets", instead, it contains one scan of data for pair-wise item co-occurrence information, and then a "Maximal Clique Enumeration" under a graph model which maps each item into a vertex and each associated item pair into an edge.

Main Properties

Compactness of association: In an association bundle there are multiple association conditions which are imposed pair-wise on bundle items, whereas in an association rule there is only one association condition which is imposed between rule antecedent and rule consequent.

Rare item problem: Different from association rule mining, association bundle identification avoids the rare item problem. The rare item problem (Mannila, 1998) refers to the mining of association rules involving low frequency items. Since lowering the support threshold to involve low frequency items may result that the number of association rules increases in a ``super-exponential" fashion, it is claimed in (Zheng, Kohavi & Mason, 2001) that "no algorithm can handle them". As such, the rare item problem and the relevant computational explosion problem become a dilemma for association rule mining. Some progresses (Han & Fu, 1995; Liu, Hsu & Ma, 1999; Tao, Murtagh & Farid, 2003; Seno & Karypis, 2005) have been made to address this dilemma. Different from association rules, association bundles impose no frequency threshold upon item(s). Therefore in association bundle identification the rare item problem disappeared - rare items can form association bundles as long as they have a strong between-item association.

Large size bundle identification: An association rule cannot have size larger than the maximum transaction size, because the simultaneous co-occurrence condition (the minimum support) is imposed on rule items. Association bundles have no minimum support requirement, thus can have large size.

FUTURE TRENDS

Association bundles have an association structure that presenting meaningful knowledge uncovered by association rules. With the similar structural analysis approach, other structures of interest can also be explored. For example, with respect to the between-set asymmetric association structure (association rules) and the between-item symmetric association structure (association bundles), the between-all-subset symmetric association structure can present a pattern that has the strongest internal association relationships. This pattern may help revealing meaningful knowledge on applications such as fraud detection, in which the fraud is detected via the identification of strongly associated behaviors. As of association bundles, research on any new pattern must be carried out over all related subjects including pattern meaning exploration, association measure design and supporting algorithm development.

CONCLUSION

In this article we describe the notion of Association Bundle Identification. Association bundles were presented by Huang et. al. (Huang, Krneta, Lin & Wu, 2006) as a new pattern of association for data mining. On applications such as the Market Basket Analysis, association bundles can be compared to, but essentially distinguished from the well-established association rules. Association bundles present meaningful and important associations that association rules unable to identify.

We describe association bundles over four aspects - association structure, association measure, threshold setting, and identification algorithms - and try to clarify these ideas via comparisons between association bundles and association rules.

REFERENCES

Agrawal, R., Imielinski, T. & Swami A. (1993). Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, 207-216.

Aggarwal, C. C. & Yu, P. S. (1998). A new framework for itemset generation. In *Proceedings of the Symposium on Principles of Database Systems*, 18-24.

Brin, S., Motwani, R., Ullman, J. D. & Shalom, T. S. (1997). Dynamic itemset counting and implication rules for market basket data. In *Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data*, 255-264.

Bruzzese, D. & Davino, C. (2001). Pruning of discovered association rules. *Computational Statistics*, *16*, 387-398.

Barber, B. & Hamilton, H. J. (2003). Extracting share frequent itemsets with infrequent subsets. *Data Mining and Knowledge Discovery*, *7*, 153-185.

Han, J. & Fu, Y. (1995). Discovery of multiple-level association rules from large databases. In *Proceedings of 1995 Int'l Conf. on Very Large Data Bases*, 420-431.

Han, J., Pei, J., Yin, Y. & Mao, R. (2004). Mining frequent patterns without candidate generation. *Data Mining and Knowledge Discovery, 8*, 53-87.

Huang, W., Krneta, M., Lin, L. & Wu, J. (2006), Association bundle – A new pattern for association analysis. In *Proceedings of the Int'l Conf. on Data Mining - Workshop on Data Mining for Design and Marketing*.

Li, J. (2006). On optimal rule discovery. *IEEE Transactions on Knowledge and Data Engineering, 18*(4), 460-471.

Liu, B., Hsu, W. & Ma, Y. (1999). Mining association rules with multiple minimum supports. In *Proceedings of the 1999 ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 337-341.

Mannila, H. (1998), Database methods for data mining, KDD-98 tutorial.

Omiecinski, E.R.(2003). Alternative interest measures for mining associations in databases. *IEEE Transactions on Knowledge and Data Engineering*, *15*(1), 57-69.

Pei, J., Han, J. & Mao, R. (2000). CLOSET: An efficient algorithm for mining frequent closed itemsets. *ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, 21-30.

Seno, M. & Karypis, G. (2005). Finding frequent itemsets using length-decreasing support constraint. *Data Mining and Knowledge Discovery*, *10*, 197-228.

Scheffer, T. (2005). Finding association rules that trade support optimally against confidence. *Intelligent Data Analysis*, *9*(4), 381-395.

Piatetsky-Shapiro, G. (1991). Discovery, Analysis, and Presentation of Strong Rules. *Knowledge Discovery in Databases*. Cambridge, MA: AAAI/MIT Press.

Tan, P., Kumar, V. & Srivastava, J. (2004). Selecting the right objective measure for association analysis. *Information Systems, 29*(4), 293-313.

Tao, F., Murtagh F. & Farid, M. (2003). Weighted association rule mining using weighted support and significance framework. In *Proceedings of the 2003 ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 661-666.

Zaki, M. J. (2000). Scalable Algorithms for Association Mining, *IEEE Transactions on Knowledge and Data Engineering,* 12(3), 372-390.

Zaki, M. J. & Hsiao, C. (2002). CHARM: An efficient algorithm for closed itemset mining. In *Proceedings*

of the Second SIAM International Conference on Data Mining, 457-473.

Zheng, Z., Kohavi, R. & Mason, L. (2001). Real world performance of association rule algorithms. In *Proceedings of the 2001 ACM SIGKDD Intentional Conference on Knowledge Discovery in Databases & Data Mining*, 401-406.

KEY TERMS AND THEIR DEFINITIONS

Association Bundle: Association bundle is a pattern that has the symmetric between-item association structure. It uses between-item lift and between-item conditional probabilities as the association measures. Its algorithm contains one scan of data set and a maximal clique enumeration problem. It can support marketing applications such as the cross-selling group of items by discounting on one of them.

Association Pattern: An association pattern describes how a group of items are statistically associated together. Association rule is an association pattern of how two sets of items are associated with each other, and association bundle is an association pattern of how individual items are pair-wise associated with each other.

Association Rule: Association rule is a pattern that has the asymmetric between-set association structure. It uses support as the rule significance measure, and confidence as the association measure. Its algorithm computes frequent itemsets. It can support marketing applications such as the shopping recommendation based on in-basket items.

Association Structure: Association structure describes the structural features of the relationships in an association pattern, such as how many relationships are contained in a pattern, whether each relationship is asymmetric or symmetric, between-set or between-item.

Asymmetric Between-set Structure: Association rules have the asymmetric between-set association structure, that is, the association relationship in an association rule exists between two sets of item and it is asymmetric from rule antecedent to rule consequent.

Strength Level Threshold: In association bundle identification, the threshold for association measure takes values in a range determined by data, which is different from the fixed range [0,1] used in association rule mining. When linearly mapping this range into [0,1], the transformed threshold is defined as the Strength Level Threshold.

Symmetric Between-item Structure: Association bundle has the symmetric between-item association structure, that is, the association relationship exists between each pair of individual items and is in a symmetric way.

Association Rule Hiding Methods

Vassilios S. Verykios
University of Thessaly, Greece

INTRODUCTION

The enormous expansion of data collection and storage facilities has created an unprecedented increase in the need for data analysis and processing power. *Data mining* has long been the catalyst for automated and sophisticated data analysis and interrogation. Recent advances in data mining and *knowledge discovery* have generated controversial impact in both scientific and technological arenas. On the one hand, data mining is capable of analyzing vast amounts of information within a minimum amount of time, an analysis that has exceeded the expectations of even the most imaginative scientists of the last decade. On the other hand, the excessive processing power of intelligent algorithms which is brought with this new research area puts at risk sensitive and confidential information that resides in large and distributed data stores.

Privacy and security risks arising from the use of data mining techniques have been first investigated in an early paper by O' Leary (1991). Clifton & Marks (1996) were the first to propose possible remedies to the protection of sensitive data and sensitive knowledge from the use of data mining. In particular, they suggested a variety of ways like the use of controlled access to the data, fuzzification of the data, elimination of unnecessary groupings in the data, data augmentation, as well as data auditing. A subsequent paper by Clifton (2000) made concrete early results in the area by demonstrating an interesting approach for privacy protection that relies on sampling. A main result of Clifton's paper was to show how to determine the right sample size of the public data (data to be disclosed to the public where sensitive information has been trimmed off), by estimating at the same time the error that is introduced from the sampling to the significance of the rules. Agrawal and Srikant (2000) were the first to establish a new research area, the *privacy preserving data mining*, which had as its goal to consider privacy and confidentiality issues originating in the mining of the data. The authors proposed an approach known as *data perturbation* that relies on disclosing a modified database with noisy data instead of the original database. The modified database could produce very similar patterns with those of the original database.

BACKGROUND

One of the main problems which have been investigated within the context of privacy preserving data mining is the so-called *association rule hiding*. Association rule hiding builds on the data mining area of *association rule mining* and studies the problem of hiding sensitive association rules from the data. The problem can be formulated as follows.

Let $I = \{i_1, i_2, \ldots, i_n\}$ be a set of binary literals, called items. Let D be a transactional database, where each transaction T contains a set of items (also called an itemset) from I, such that $T \subseteq I$. A unique identifier TID (stands for transaction id) is associated with each transaction. We assume that the items in an itemset are sorted in lexicographic order. An *association rule* is an implication of the form $X \Rightarrow Y$, where $X \subset I$, $Y \subset I$ and $X \cap Y = \varnothing$. We say that a rule $X \Rightarrow Y$ holds in the database D with *confidence* c if $|X \cup Y|/|X| \geq c$ (where $|X|$ is the cardinality of the set X) and *support* s if $|X \cup Y|/N \geq s$, where N is the number of transactions in D. An association rule mining algorithm proceeds by finding all itemsets that appear frequently enough in the database, so that they can be considered interesting, and by deriving from them all proper association rules that are strong (above a lower confidence level) enough. The association rule hiding problem aims at the prevention of a subset of the association rules from being disclosed during mining. We call these rules *sensitive*, and we argue that in order for a rule to become non-sensitive, its support and confidence must be brought below the minimum support and confidence threshold, so that it escapes mining at the corresponding levels of support and confidence. More formally we can state: Given a database D, a set R of rules mined from database D at a pre-specified threshold of support and confidence, and a subset R_h ($R_h \subset R$) of sensitive rules, the association

rule hiding refers to transforming the database D into a database D' of the same degree (same number of items) as D in such a way that only the rules in R-R_h can be mined from D' at either the pre-specified or even higher thresholds. We should note here that in the association rule hiding problem we consider the publishing of a modified database instead of the secure rules because we claim that a modified database will certainly have higher utility to the data holder compared to the set of secure rules. This claim relies on the fact that either a different data mining approach may be applied to the published data, or a different support and confidence threshold may be easily selected by the data miner, if the data itself is published.

It has been proved (Atallah, Bertino, Elmagarmid, Ibrahim, & Verykios, 1999) that the association rule hiding problem which is also referred to as the *database sanitization problem* is NP-hard. Towards the solution of this problem a number of heuristic and exact techniques have been introduced. In the following section we present a thorough analysis of some of the most interesting techniques which have been proposed for the solution of the association rule hiding problem.

MAIN FOCUS

In the following discussion we present three classes of state of the art techniques which have been proposed for the solution of the association rule hiding problem. The first class contains the *perturbation* approaches which rely on heuristics for modifying the database values so that the sensitive knowledge is hidden. The *use of unknowns* for the hiding of rules comprises the second class of techniques to be investigated in this expository study. The third class contains recent sophisticated approaches that provide a new perspective to the association rule hiding problem, as well as a special class of computationally expensive solutions, the *exact solutions*.

Perturbation Approaches

Atallah, Bertino, Elmagarmid, Ibrahim & Verykios (1999) were the first to propose a rigorous solution to the association rule hiding problem. Their approach was based on the idea of preventing disclosure of sensitive rules by decreasing the support of the itemsets generating the sensitive association rules. This reduced hiding approach is also known as frequent itemset hiding. The heuristic employed in their approach traverses the itemset lattice in the space of items from bottom to top in order to identify these items that need to turn from 1 to 0 so that the support of an itemset that corresponds to a sensitive rule becomes lower than the minimum support threshold. The algorithm sorts the sensitive itemsets based on their supports and then it proceeds by hiding all of the sensitive itemsets one by one. A major improvement over the first heuristic algorithm which was proposed in the previous work appeared in the work of Dasseni, Verykios, Elmagarmid & Bertino (2001). The authors extended the existing association rule hiding technique from using only the support of the generating frequent itemsets to using both the support of the generating frequent itemsets and the confidence of the association rules. In that respect, they proposed three new algorithms that exhibited interesting behavior with respect to the characteristics of the hiding process. Verykios, Elmagarmid, Bertino, Saygin & Dasseni (2004) along the same lines of the first work, presented five different algorithms based on various hiding strategies, and they performed an extensive evaluation of these algorithms with respect to different metrics like the execution time, the number of changes in the original data, the number of non-sensitive rules which were hidden (hiding side effects or false rules) and the number of "ghost" rules which were produced after the hiding. Oliveira & Zaiane (2002) extended existing work by focusing on algorithms that solely remove information so that they create a smaller impact in the database by not generating false or ghost rules. In their work they considered two classes of approaches: the pattern restriction based approaches that remove patterns completely from sensitive transactions, and the item restriction based approaches that selectively remove items from sensitive transactions. They also proposed various performance measures for quantifying the fraction of mining patterns which are preserved after sanitization.

Use of Unknowns

A completely different approach to the hiding of sensitive association rules was taken by employing the use of unknowns in the hiding process (Saygin, Verykios & Elmagarmid, 2002, Saygin, Verykios & Clifton, 2001). The goal of the algorithms that incorporate unknowns in the hiding process was to obscure a given

set of sensitive rules by replacing known values by unknowns, while minimizing the side effects on non-sensitive rules. Note here that the use of unknowns needs a high level of sophistication in order to perform equally well as the perturbation approaches that we presented before, although the quality of the datasets after hiding is higher than that in the perturbation approaches since values do not change behind the scene. Although the work presented under this category is in an early stage, the authors do give arguments as to the difficulty of recovering sensitive rules as well as they formulate experiments that test the side effects on non-sensitive rules. Among the new ideas which were proposed in this work, is the modification of the basic notions of support and confidence in order to accommodate for the use of unknowns (think how an unknown value will count during the computation of these two metrics) and the introduction of a new parameter, the *safety margin*, which was employed in order to account for the distance below the support or the confidence threshold that a sensitive rule needs to maintain. Further studies related to the use of unknown values for the hiding of sensitive rules are underway (Wang & Jafari, 2005).

Recent Approaches

The problem of inverse frequent itemset mining was defined by Mielikainen (2003) in order to answer the following research problem: Given a collection of frequent itemsets and their support, find a transactional database such that the new database precisely agrees with the supports of the given frequent itemset collection while the supports of other itemsets would be less than the predetermined threshold. A recent study (Chen, Orlowska & Li, 2004) investigates the problem of using the concept of inverse frequent itemset mining to solve the association rule hiding problem. In particular, the authors start from a database on which they apply association rule mining. After the association rules have been mined and organized into an itemset lattice, the lattice is revised by taking into consideration the sensitive rules. This means that the frequent itemsets that have generated the sensitive rules are forced to become infrequent in the lattice. Given the itemsets that remain frequent in the lattice after the hiding of the sensitive itemsets, the proposed algorithm tries to reconstruct a new database, the mining of which will produce the given frequent itemsets.

Another study (Menon, Sarkar & Mukherjee 2005) was the first to formulate the association rule hiding problem as an integer programming task by taking into account the occurrences of sensitive itemsets in the transactions. The solution of the integer programming problem provides an answer as to the minimum number of transactions that need to be sanitized for each sensitive itemset to become hidden. Based on the integer programming solution, two heuristic approaches are presented for actually identifying the items to be sanitized.

A border based approach along with a hiding algorithm is presented in Sun & Yu (2005). The authors propose the use of the border of frequent itemsets to drive the hiding algorithm. In particular, given a set of sensitive frequent itemsets, they compute the new (revised) border on which the sensitive itemsets have just turned to infrequent. In this way, the hiding algorithm is forced to maintain the itemsets in the revised positive border while is trying to hide those itemsets in the negative border which have moved from frequent to infrequent. A *maxmin* approach (Moustakides & Verykios, 2006) is proposed that relies on the border revision theory by using the maxmin criterion which is a method in decision theory for maximizing the minimum gain. The maxmin approach improves over the basic border based approach both in attaining hiding results of better quality and in achieving much lower execution times. An exact approach (Gkoulalas-Divanis & Verykios 2006) that is also based on the border revision theory relies on an integer programming formulation of the hiding problem that is efficiently solved by using a Binary Integer Programming approach. The important characteristic of the exact solutions is that they do not create any hiding side effects.

Wu, Chiang & Chen (2007) present a limited side effect approach that modifies the original database to hide sensitive rules by decreasing their support or confidence. The proposed approach first classifies all the valid modifications that can affect the sensitive rules, the non-sensitive rules, and the spurious rules. Then, it uses heuristic methods to modify the transactions in an order that increases the number of hidden sensitive rules, while reducing the number of modified entries. Amiri (2007) presents three data sanitization heuristics that demonstrate high data utility at the expense of computational speed. The first heuristic reduces the support of the sensitive itemsets by deleting a set of supporting transactions. The second heuristic modifies, instead of

deleting, the supporting transactions by removing some items until the sensitive itemsets are protected. The third heuristic combines the previous two by using the first approach to identify the sensitive transactions and the second one to remove items from these transactions, until the sensitive knowledge is hidden.

Still another approach (Oliveira, Zaiane & Saygin, 2004) investigates the distribution of non-sensitive rules for security reasons instead of publishing the perturbed database. The proposed approach presents a rule sanitization algorithm for blocking inference channels that may lead to the disclosure of sensitive rules.

FUTURE TRENDS

Many open issues related to the association rule hiding problem are still under investigation. The emergence of sophisticated exact hiding approaches of extremely high complexity, especially for very large databases, causes the consideration of efficient parallel approaches to be employed for the solution of this problem. A lot more work is in need to provide hiding solutions that take advantage of the use of unknowns. More sophisticated techniques need to emerge regarding the solution of the hiding problem by making use of database reconstruction approaches. Ongoing work considers yet another solution which is to append to the original database a synthetically generated database so that the sensitive knowledge is hidden in the combined database which is disclosed to the public.

CONCLUSION

In the information era, privacy comprises one of the most important issues that need to be thoroughly investigated and resolved before data and information can be given to the public to serve different goals. Privacy is not constrained to personally identifiable information, but it can be equally well refer to business information or other forms of knowledge which can be produced from the processing of the data through data mining and knowledge discovery approaches. The problem of association rule hiding has been in the forefront of the privacy preserving data mining area for more than a decade now. Recently proposed approaches have been creating enormous impact in the area while at the same time open the way to new research problems. Although the systematic work all these years have created a lot of research results, there is still a lot of work to be done. Apart from ongoing work in the field we foresee the need of applying these techniques to operational data warehouses so that we can evaluate in a real environment their effectiveness and applicability. We also envision the necessity of applying knowledge hiding techniques to distributed environments where information and knowledge is shared among collaborators and/or competitors.

REFERENCES

Agrawal, R., & Srikant, R.(2000). Privacy-Preserving Data Mining. *SIGMOD Conference,* 439-450.

Amiri, A. (2007). Dare to share: Protecting Sensitive Knowledge with Data Sanitization. *Decision Support Systems*, 43(1), 181-191.

Atallah, M., Bertino, E., Elmagarmid, A.K., Ibrahim, M., & Verykios, V.S. (1999). Disclosure Limitation of Sensitive Rules. *IEEE Knowledge and Data Engineering Exchange Workshop*, 45-52.

Chen, X., Orlowska, M., & Li, X. (2004). A New Framework for Privacy Preserving Data Sharing, *Privacy and Security Aspects of Data Mining Workshop*, 47-56.

Clifton, C. (2000). Using Sample Size to Limit Exposure to Data Mining. *Journal of Computer Security*, 8(4), 281-307.

Dasseni, E., Verykios, V.S., Elmagarmid, A.K., & Bertino, E. (2000). Hiding Association Rules by Using Confidence and Support. *Information Hiding*, 369-383

Gkoulalas-Divanis, A., & Verykios, V.S. (2006). An integer programming approach for frequent itemset hiding. *CIKM*, 748-757

Mielikainen, T. (2003). On inverse frequent set mining. In Wenliang Du and Chris Clifton (Eds.): *Proceedings of the 2nd Workshop on Privacy Preserving Data Mining*, 18-23. IEEE Computer Society.

Menon, S., Sarkar, S., & Mukherjee, S. (2005). Maximizing Accuracy of Shared Databases when Concealing Sensitive Patterns. *Information Systems Research*, 16(3), 256-270.

Moustakides, G.V., & Verykios, V.S. (2006). A Max-Min Approach for Hiding Frequent Itemsets. *ICDM Workshops*, 502-506.

Oliveira, S.R.M., & Zaïane, O.R. (2003). Algorithms for Balancing Privacy and Knowledge Discovery in Association Rule Mining. *IDEAS*, 54-65.

O'Leary, D.E. (1991). Knowledge Discovery as a Threat to Database Security. *Knowledge Discovery in Databases*, 507-516.

Oliveira, S.R.M., Zaïane, O.R., & Saygin, Y. (2004). Secure Association Rule Sharing. *PAKDD*: 74-85.

Saygin, Y., Verykios, V.S., & Clifton, C. (2001). Using Unknowns to Prevent Discovery of Association Rules. *SIGMOD Record* 30(4), 45-54.

Saygin, Y., Verykios, V.S., & Elmagarmid, A.K. (2002). Privacy Preserving Association Rule Mining. *RIDE,* 151-158.

Sun, X., & Yu, P.S. (2005). A Border-Based Approach for Hiding Sensitive Frequent Itemsets. *ICDM*, 426-433.

Verykios, V.S., Elmagarmid, A.K., Bertino, E., Saygin, Y., & Dasseni, E. (2004): Association Rule Hiding. *IEEE Trans. Knowl. Data Eng.* 16(4), 434-447.

Wang, S.L., & Jafari, A. (2005). Using unknowns for hiding sensitive predictive association rules. *IRI*, 223-228.

Wu, Y.H., Chiang C.M., & Chen A.L.P. (2007) Hiding Sensitive Association Rules with Limited Side Effects. *IEEE Trans. Knwol. Data Eng. 1*9(1), 29-42.

KEY TERMS

Association Rule Hiding: The process of lowering the interestingness of an association rule in the database by either decreasing the support or the confidence of the rule, while the number of changes and side effects is minimized.

Database Reconstruction: Generation of a database that exhibits certain statistical behavior. Such a database can be used instead of the original database (i.e., be disclosed to the public) with the added value that privacy is not breached.

Data Sanitization: The process of removing sensitive information from the data, so that they can be made public.

Exact Hiding Approaches: Hiding approaches that provide solutions without side effects, if such solutions exist.

Frequent Itemset Hiding: The process of decreasing the support of a frequent itemset in the database by decreasing the support of individual items that appear in this frequent itemset.

Heuristic Hiding Approaches: Hiding approaches that rely on heuristics in order to become more efficient. These approaches usually behave sub-optimally with respect to the side effects that they create.

Inverse Frequent Itemset Mining: Given a set of frequent itemsets along with their supports, the task of the inverse frequent itemset mining problem is to construct the database which produces the specific set of frequent itemsets as output after mining.

Knowledge Hiding: The process of hiding sensitive knowledge in the data. This knowledge can be in a form that can be mined from a data warehouse through a data mining algorithm in a knowledge discovery in databases setting like association rules, a classification or clustering model, a summarization model etc.

Perturbed Database: A hiding algorithm modifies the original database so that sensitive knowledge (itemsets or rules) is hidden. The modified database is known as perturbed database.

Privacy Preserving Data Mining: The subfield of data mining that is investigating various issues related to the privacy of information and knowledge during the mining of the data.

Sensitive Itemset: A security administrator determines the sensitivity level of a frequent itemset. A frequent itemset that is found above a certain sensitivity level is considered as sensitive. Sensitive itemsets need to be protected by hiding techniques.

Association Rule Mining

Yew-Kwong Woon
Nanyang Technological University, Singapore

Wee-Keong Ng
Nanyang Technological University, Singapore

Ee-Peng Lim
Nanyang Technological University, Singapore

INTRODUCTION

Association Rule Mining (ARM) is concerned with how items in a transactional database are grouped together. It is commonly known as market basket analysis, because it can be likened to the analysis of items that are frequently put together in a basket by shoppers in a market. From a statistical point of view, it is a semiautomatic technique to discover correlations among a set of variables.

ARM is widely used in myriad applications, including recommender systems (Lawrence, Almasi, Kotlyar, Viveros, & Duri, 2001), promotional bundling (Wang, Zhou, & Han, 2002), Customer Relationship Management (CRM) (Elliott, Scionti, & Page, 2003), and cross-selling (Brijs, Swinnen, Vanhoof, & Wets, 1999). In addition, its concepts have also been integrated into other mining tasks, such as Web usage mining (Woon, Ng, & Lim, 2002), clustering (Yiu & Mamoulis, 2003), outlier detection (Woon, Li, Ng, & Lu, 2003), and classification (Dong & Li, 1999), for improved efficiency and effectiveness.

CRM benefits greatly from ARM as it helps in the understanding of customer behavior (Elliott et al., 2003). Marketing managers can use association rules of products to develop joint marketing campaigns to acquire new customers. The application of ARM for the cross-selling of supermarket products has been successfully attempted in many cases (Brijs et al., 1999). In one particular study involving the personalization of supermarket product recommendations, ARM has been applied with much success (Lawrence et al., 2001). Together with customer segmentation, ARM helped to increase revenue by 1.8%.

In the biology domain, ARM is used to extract novel knowledge on protein-protein interactions (Oyama, Kitano, Satou, & Ito, 2002). It is also successfully applied in gene expression analysis to discover biologically relevant associations between different genes or between different environment conditions (Creighton & Hanash, 2003).

BACKGROUND

Recently, a new class of problems emerged to challenge ARM researchers: Incoming data is streaming in too fast and changing too rapidly in an unordered and unbounded manner. This new phenomenon is termed data stream (Babcock, Babu, Datar, Motwani, & Widom, 2002).

One major area where the data stream phenomenon is prevalent is the World Wide Web (Web). A good example is an online bookstore, where customers can purchase books from all over the world at any time. As a result, its transactional database grows at a fast rate and presents a scalability problem for ARM. Traditional ARM algorithms, such as Apriori, were not designed to handle large databases that change frequently (Agrawal & Srikant, 1994). Each time a new transaction arrives, Apriori needs to be restarted from scratch to perform ARM. Hence, it is clear that in order to conduct ARM on the latest state of the database in a timely manner, an incremental mechanism to take into consideration the latest transaction must be in place.

In fact, a host of incremental algorithms have already been introduced to mine association rules incrementally (Sarda & Srinivas, 1998). However, they are only incremental to a certain extent; the moment the universal itemset (the number of unique items in a database) (Woon, Ng, & Das, 2001) is changed, they have to be restarted from scratch. The universal

itemset of any online store would certainly be changed frequently, because the store needs to introduce new products and retire old ones for competitiveness. Moreover, such incremental ARM algorithms are efficient only when the database has not changed much since the last mining.

The use of data structures in ARM, particularly the trie, is one viable way to address the data stream phenomenon. Data structures first appeared when programming became increasingly complex during the 1960s. In his classic book, *The Art of Computer Programming* Knuth (1968) reviewed and analyzed algorithms and data structures that are necessary for program efficiency. Since then, the traditional data structures have been extended, and new algorithms have been introduced for them. Though computing power has increased tremendously over the years, efficient algorithms with customized data structures are still necessary to obtain timely and accurate results. This fact is especially true for ARM, which is a computationally intensive process.

The trie is a multiway tree structure that allows fast searches over string data. In addition, as strings with common prefixes share the same nodes, storage space is better utilized. This makes the trie very useful for storing large dictionaries of English words. Figure 1 shows a trie storing four English words *(ape, apple, base,* and *ball).* Several novel trielike data structures have been introduced to improve the efficiency of ARM, and we discuss them in this section.

Amir, Feldman, & Kashi (1999) presented a new way of mining association rules by using a trie to preprocess the database. In this approach, all transactions are mapped onto a trie structure. This mapping involves the extraction of the powerset of the transaction items and the updating of the trie structure. Once built, there is no longer a need to scan the database to obtain support counts of itemsets, because the trie structure contains all their support counts. To find frequent itemsets, the structure is traversed by using depth-first search, and itemsets with support counts satisfying the minimum *support threshold* are added to the set of frequent itemsets.

Drawing upon that work, Yang, Johar, Grama, & Szpankowski (2000) introduced a binary *Patricia trie* to reduce the heavy memory requirements of the preprocessing trie. To support faster support queries, the authors added a set of horizontal pointers to index nodes. They also advocated the use of some form of primary threshold to further prune the structure. However, the compression achieved by the compact Patricia trie comes at a hefty price: It greatly complicates the horizontal pointer index, which is a severe overhead. In addition, after compression, it will be difficult for the Patricia trie to be updated whenever the database is altered.

The *Frequent Pattern-growth* (FP-growth) algorithm is a recent association rule mining algorithm that achieves impressive results (Han, Pei, Yin, & Mao, 2004). It uses a compact tree structure called a *Frequent Pattern-tree* (FP-tree) to store information about frequent 1-itemsets. This compact structure removes the need for multiple database scans and is constructed with only 2 scans. In the first database scan, frequent 1-itemsets are obtained and sorted in support descending order. In the second scan, items in the transactions are first sorted according to the order of the frequent 1-itemsets. These sorted items are used to construct the FP-tree. Figure 2 shows an FP-tree constructed from the database in Table 1.

FP-growth then proceeds to recursively mine FP-trees of decreasing size to generate frequent itemsets

Figure 1. An example of a trie for storing English words

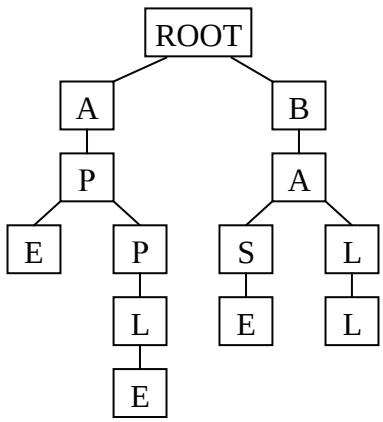


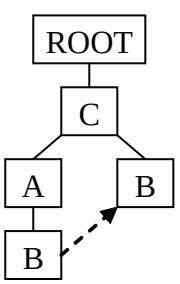
Table 1. A sample transactional database

TID	Items
100	*AC*
200	*BC*
300	*ABC*
400	*ABCD*

without candidate generation and database scans. It does so by examining all the *conditional pattern bases* of the FP-tree, which consists of the set of frequent itemsets occurring with the suffix pattern. Conditional FP-trees are constructed from these conditional pattern bases, and mining is carried out recursively with such trees to discover frequent itemsets of various sizes. However, because both the construction and the use of the FP-trees are complex, the performance of FP-growth is reduced to be on par with Apriori at support thresholds of 3% and above. It only achieves significant speed-ups at support thresholds of 1.5% and below. Moreover, it is only incremental to a certain extent, depending on the FP-tree *watermark* (validity support threshold). As new transactions arrive, the support counts of items increase, but their relative support frequency may decrease, too. Suppose, however, that the new transactions cause too many previously infrequent itemsets to become frequent — that is, the watermark is raised too high (in order to make such itemsets infrequent) according to a user-defined level — then the FP-tree must be reconstructed.

The use of lattice theory in ARM was pioneered by Zaki (2000). Lattice theory allows the vast search space to be decomposed into smaller segments that can be tackled independently in memory or even in other machines, thus promoting parallelism. However, they require additional storage space as well as different traversal and construction techniques. To complement the use of lattices, Zaki uses a vertical database format, where each itemset is associated with a list of transactions known as a tid-list (transaction identifier–list). This format is useful for fast frequency counting of itemsets but generates additional overheads because most databases have a horizontal format and would need to be converted first.

Figure 2. An FP-tree constructed from the database in Table 1 at a support threshold of 50%



The Continuous Association Rule Mining Algorithm (CARMA), together with the support lattice, allows the user to change the support threshold and continuously displays the resulting association rules with support and confidence bounds during its first scan/phase (Hidber, 1999). During the second phase, it determines the precise support of each itemset and extracts all the frequent itemsets. CARMA can readily compute frequent itemsets for varying support thresholds. However, experiments reveal that CARMA only performs faster than Apriori at support thresholds of 0.25% and below, because of the tremendous overheads involved in constructing the support lattice.

The adjacency lattice, introduced by Aggarwal & Yu (2001), is similar to Zaki's boolean powerset lattice, except the authors introduced the notion of adjacency among itemsets, and it does not rely on a vertical database format. Two itemsets are said to be adjacent to each other if one of them can be transformed to the other with the addition of a single item. To address the problem of heavy memory requirements, a primary threshold is defined. This term signifies the minimum support threshold possible to fit all the qualified itemsets into the adjacency lattice in main memory. However, this approach disallows the mining of frequent itemsets at support thresholds lower than the primary threshold.

MAIN THRUST

As shown in our previous discussion, none of the existing data structures can effectively address the issues induced by the data stream phenomenon. Here are the desirable characteristics of an ideal data structure that can help ARM cope with data streams:

- It is highly scalable with respect to the size of both the database and the universal itemset.
- It is incrementally updated as transactions are added or deleted.
- It is constructed independent of the support threshold and thus can be used for various support thresholds.
- It helps to speed up ARM algorithms to a certain extent that allows results to be obtained in real-time.

We shall now discuss our novel trie data structure that not only satisfies the above requirements but

also outperforms the discussed existing structures in terms of efficiency, effectiveness, and practicality. Our structure is termed Support-Ordered Trie Itemset (SOTrieIT — pronounced "so-try-it"). It is a dual-level support-ordered trie data structure used to store pertinent itemset information to speed up the discovery of frequent itemsets.

As its construction is carried out before actual mining, it can be viewed as a preprocessing step. For every transaction that arrives, 1-itemsets and 2-itemsets are first extracted from it. For each itemset, the SOTrieIT will be traversed in order to locate the node that stores its support count. Support counts of 1-itemsets and 2-itemsets are stored in first-level and second-level nodes, respectively. The traversal of the SOTrieIT thus requires at most two redirections, which makes it very fast. At any point in time, the SOTrieIT contains the support counts of all 1-itemsets and 2-itemsets that appear in all the transactions. It will then be sorted level-wise from left to right according to the support counts of the nodes in descending order.

Figure 3 shows a SOTrieIT constructed from the database in Table 1. The bracketed number beside an item is its support count. Hence, the support count of itemset {AB} is 2. Notice that the nodes are ordered by support counts in a level-wise descending order.

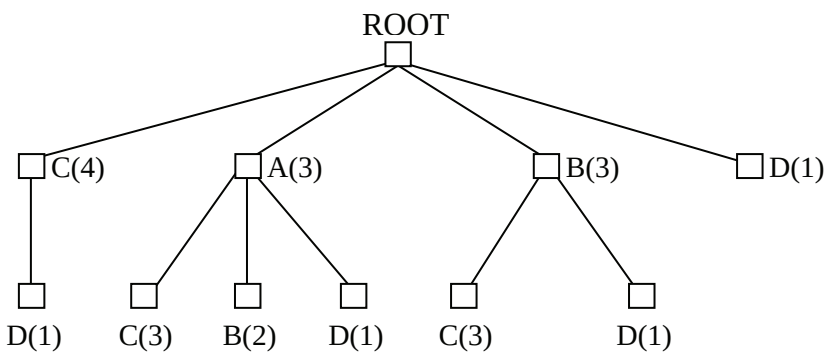
In algorithms such as FP-growth that use a similar data structure to store itemset information, the structure must be rebuilt to accommodate updates to the universal itemset. The SOTrieIT can be easily updated to accommodate the new changes. If a node for a new item in the universal itemset does not exist, it will be created and inserted into the SOTrieIT accordingly. If an item is removed from the universal itemset, all nodes containing that item need only be removed, and the rest of the nodes would still be valid.

Unlike the trie structure of Amir et al. (1999), the SOTrieIT is ordered by support count (which speeds up mining) and does not require the powersets of transactions (which reduces construction time). The main weakness of the SOTrieIT is that it can only discover frequent 1-itemsets and 2-itemsets; its main strength is its speed in discovering them. They can be found promptly because there is no need to scan the database. In addition, the search (depth first) can be stopped at a particular level the moment a node representing a nonfrequent itemset is found, because the nodes are all support ordered.

Another advantage of the SOTrieIT, compared with all previously discussed structures, is that it can be constructed *online*, meaning that each time a new transaction arrives, the SOTrieIT can be incrementally updated. This feature is possible because the SOTrieIT is constructed without the need to know the support threshold; it is support independent. All 1-itemsets and 2-itemsets in the database are used to update the SOTrieIT regardless of their support counts. To conserve storage space, existing trie structures such as the FP-tree have to use thresholds to keep their sizes manageable; thus, when new transactions arrive, they have to be reconstructed, because the support counts of itemsets will have changed.

Finally, the SOTrieIT requires far less storage space than a trie or Patricia trie because it is only two levels deep and can be easily stored in both memory and files. Although this causes some input/output (I/O) overheads, it is insignificant as shown in our extensive experiments. We have designed several algorithms to

Figure 3. A SOTrieIT structure



work synergistically with the SOTrieIT and, through experiments with existing prominent algorithms and a variety of databases, we have proven the practicality and superiority of our approach (Das, Ng, & Woon, 2001; Woon et al., 2001). In fact, our latest algorithm, *FOLD-growth,* is shown to outperform FP-growth by more than 100 times (Woon, Ng, & Lim, 2004).

FUTURE TRENDS

The data stream phenomenon will eventually become ubiquitous as Internet access and bandwidth become increasingly affordable. With keen competition, products will become more complex with customization and more varied to cater to a broad customer base; transaction databases will grow in both size and complexity. Hence, association rule mining research will certainly continue to receive much attention in the quest for faster, more scalable and more configurable algorithms.

CONCLUSION

Association rule mining is an important data mining task with several applications. However, to cope with the current explosion of raw data, data structures must be utilized to enhance its efficiency. We have analyzed several existing trie data structures used in association rule mining and presented our novel trie structure, which has been proven to be most useful and practical. What lies ahead is the parallelization of our structure to further accommodate the ever-increasing demands of today's need for speed and scalability to obtain association rules in a timely manner. Another challenge is to design new data structures that facilitate the discovery of trends as association rules *evolve* over time. Different association rules may be mined at different time points and, by understanding the patterns of changing rules, additional interesting knowledge may be discovered.

REFERENCES

Aggarwal, C. C., & Yu, P. S. (2001). A new approach to online generation of association rules. *IEEE Transactions on Knowledge and Data Engineering, 13*(4), 527-540.

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules. *Proceedings of the 20th International Conference on Very Large Databases* (pp. 487-499), Chile.

Amir, A., Feldman, R., & Kashi, R. (1999). A new and versatile method for association generation. *Information Systems, 22*(6), 333-347.

Babcock, B., Babu, S., Datar, M., Motwani, R., & Widom, J. (2002). Models and issues in data stream systems. *Proceedings of the ACM SIGMOD/PODS Conference* (pp. 1-16), USA.

Brijs, T., Swinnen, G., Vanhoof, K., & Wets, G. (1999). Using association rules for product assortment decisions: A case study. *Proceedings of the Fifth ACM SIGKDD Conference* (pp. 254-260), USA.

Creighton, C., & Hanash, S. (2003). Mining gene expression databases for association rules. *Bioinformatics, 19*(1), 79-86.

Das, A., Ng, W. K., & Woon, Y. K. (2001). Rapid association rule mining. *Proceedings of the 10th International Conference on Information and Knowledge Management* (pp. 474-481), USA.

Dong, G., & Li, J. (1999). Efficient mining of emerging patterns: Discovering trends and differences. *Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining* (pp. 43-52), USA.

Elliott, K., Scionti, R., & Page, M. (2003). *The confluence of data mining and market research for smarter CRM*. Retrieved from *http://www.spss.com/home_page/wp133.htm*

Han, J., Pei, J., Yin Y., & Mao, R. (2004). Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data Mining and Knowledge Discovery, 8*(1), 53-97.

Hidber, C. (1999). Online association rule mining. *Proceedings of the ACM SIGMOD Conference* (pp. 145-154), USA.

Knuth, D.E. (1968). The art of computer programming, Vol. 1. *Fundamental Algorithms.* Addison-Wesley Publishing Company.

Lawrence, R. D., Almasi, G. S., Kotlyar, V., Viveros, M. S., & Duri, S. (2001). Personalization of supermarket

product recommendations. *Data Mining and Knowledge Discovery, 5*(1/2), 11-32.

Oyama, T., Kitano, K., Satou, K., & Ito, T. (2002). Extraction of knowledge on protein-protein interaction by association rule discovery. *Bioinformatics, 18*(5), 705-714.

Sarda, N. L., & Srinivas, N. V. (1998). An adaptive algorithm for incremental mining of association rules. *Proceedings of the Ninth International Conference on Database and Expert Systems* (pp. 240-245), Austria.

Wang, K., Zhou, S., & Han, J. (2002). Profit mining: From patterns to actions. *Proceedings of the Eighth International Conference on Extending Database Technology* (pp. 70-87), Prague.

Woon, Y. K., Li, X., Ng, W. K., & Lu, W. F. (2003). Parameterless data compression and noise filtering using association rule mining. *Proceedings of the Fifth International Conference on Data Warehousing and Knowledge Discovery* (pp. 278-287), Prague.

Woon, Y. K., Ng, W. K., & Das, A. (2001). Fast online dynamic association rule mining. *Proceedings of the Second International Conference on Web Information Systems Engineering* (pp. 278-287), Japan.

Woon, Y. K., Ng, W. K., & Lim, E. P. (2002). Online and incremental mining of separately grouped web access logs. *Proceedings of the Third International Conference on Web Information Systems Engineering* (pp. 53-62), Singapore.

Woon, Y. K., Ng, W. K., & Lim, E. P. (2004). A support-ordered trie for fast frequent itemset discovery. *IEEE Transactions on Knowledge and Data Engineering, 16*(5).

Yang, D. Y., Johar, A., Grama, A., & Szpankowski, W. (2000). Summary structures for frequency queries on large transaction sets. *Proceedings of the Data Compression Conference* (pp. 420-429).

Yiu, M. L., & Mamoulis, N. (2003). Frequent-pattern based iterative projected clustering. *Proceedings of the Third International Conference on Data Mining*, USA.

Zaki, M. J. (2000). Scalable algorithms for association mining. *IEEE Transactions on Knowledge and Data Engineering, 12*(3), 372-390.

KEY TERMS

Apriori: A classic algorithm that popularized association rule mining. It pioneered a method to generate candidate itemsets by using only frequent itemsets in the previous pass. The idea rests on the fact that any subset of a frequent itemset must be frequent as well. This idea is also known as the *downward closure* property.

Itemset: An unordered set of unique items, which may be products or features. For computational efficiency, the items are often represented by integers. A *frequent* itemset is one with a support count that exceeds the support threshold, and a *candidate* itemset is a potential frequent itemset. A *k*-itemset is an itemset with exactly *k* items.

Key: A unique sequence of values that defines the location of a node in a tree data structure.

Patricia Trie: A compressed binary trie. The *Patricia* (Practical Algorithm to Retrieve Information Coded in Alphanumeric) trie is compressed by avoiding one-way branches. This is accomplished by including in each node the number of bits to skip over before making the next branching decision.

SOTrieIT: A dual-level trie whose nodes represent itemsets. The position of a node is ordered by the support count of the itemset it represents; the most frequent itemsets are found on the leftmost branches of the SOTrieIT.

Support Count of an Itemset: The number of transactions that contain a particular itemset.

Support Threshold: A threshold value that is used to decide if an itemset is interesting/frequent. It is defined by the user, and generally, an association rule mining algorithm has to be executed many times before this value can be well adjusted to yield the desired results.

Trie: An *n*-ary tree whose organization is based on *key space* decomposition. In key space decomposition, the key range is equally subdivided, and the splitting position within the key range for each node is predefined.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 59-64, copyright 2005 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

On Association Rule Mining for the QSAR Problem

Luminita Dumitriu
"Dunarea de Jos" University, Romania

Cristina Segal
"Dunarea de Jos" University, Romania

Marian Craciun
"Dunarea de Jos" University, Romania

Adina Cocu
"Dunarea de Jos" University, Romania

INTRODUCTION

The concept of Quantitative Structure-Activity Relationship (QSAR), introduced by Hansch and co-workers in the 1960s, attempts to discover the relationship between the structure and the activity of chemical compounds (SAR), in order to allow the prediction of the activity of new compounds based on knowledge of their chemical structure alone. These predictions can be achieved by quantifying the SAR.

Initially, statistical methods have been applied to solve the QSAR problem. For example, pattern recognition techniques facilitate data dimension reduction and transformation techniques from multiple experiments to the underlying patterns of information. Partial least squares (PLS) is used for performing the same operations on the target properties. The predictive ability of this method can be tested using cross-validation on the test set of compounds.

Later, data mining techniques have been considered for this prediction problem. Among data mining techniques, the most popular ones are based on neural networks (Wang, Durst, Eberhart, Boyd, & Ben-Miled, 2004) or on neuro-fuzzy approaches (Neagu, Benfenati, Gini, Mazzatorta, & Roncaglioni, 2002) or on genetic programming (Langdon, &Barrett, 2004). All these approaches predict the activity of a chemical compound, without being able to explain the predicted value.

In order to increase the understanding on the prediction process, descriptive data mining techniques have started to be used related to the QSAR problem. These techniques are based on association rule mining.

In this chapter, we describe the use of association rule-based approaches related to the QSAR problem.

BACKGROUND

Association rule mining, introduced by (Agrawal, Imielinski & Swami, 1993), is defined as finding all the association rules between sets of items in a database that hold with more that a user-given minimum support threshold and a user-given minimum confidence threshold. According to (Agrawal, Imielinski & Swami, 1993) this problem is solved in two steps:

1. Finding all frequent itemsets in the database.
2. For each frequent itemset I, generating all association rules $I' \Rightarrow I \setminus I'$, where $I' \subset I$.

The second problem can be solved in a straightforward manner after the first step is completed. Hence, the problem of mining association rules is reduced to the problem of finding all frequent itemsets. This is not a trivial problem, since the number of possible frequent itemsets is equal to the size of the power set of I, $2^{|I|}$.

There are many algorithms proposed in the literature, most of them based on the Apriori mining method (Agrawal & Srikant, 1994) that relies on a basic property of frequent itemsets: all subsets of a frequent itemset are frequent. This property can also be stated as all supersets of an infrequent itemset are infrequent. There are other approaches, namely the closed-itemset approaches, as Close (Pasquier, Bastide, Taouil & Lakhal, 1999),

CHARM (Zaki & Hsiao, 1999) and Closet (Pei, Han & Mao, 2000). The closed-itemset approaches rely on the application of Formal Concept Analysis to association rule problem that was first mentioned in (Zaki & Ogihara, 1998). For more details on lattice theory see (Ganter & Wille, 1999). Another approach leading to a small number of results is finding representative association rules (Kryszkiewicz, 1998).

The difference between Apriori-based and closed itemset-based approaches consists in the treatment of sub-unitary confidence and unitary confidence association rules, namely Apriori makes no distinction between them, while FCA-based approaches report sub-unitary association rules (also named partial implication rules) structured in a concept lattice and, eventually, the pseudo-intents, a base on the unitary association rules (also named global implications, exhibiting a logical implication behavior). The advantage of a closed itemset approach is the smaller size of the resulting concept lattice versus the number of frequent itemsets, *i.e.* search space reduction.

MAIN THRUST OF THE CHAPTER

While there are many application domains for the association rule mining methods, they have only started to be used in relation to the QSAR problem. There are two main approaches: one that attempts classifying chemical compounds, using frequent sub-structure mining (Deshpande, Kuramochi, Wale, & Karypis, 2005), a modified version of association rule mining, and one that attempts predicting activity using an association rule-based model (Dumitriu, Segal, Craciun, Cocu, & Georgescu, 2006).

Mined Data

For the QSAR problem, the items are called chemical compound descriptors. There are various types of descriptors that can be used to represent the chemical structure of compounds: chemical element presence in a compound, chemical element mass, normalized chemical element mass, topological structure of the molecule, geometrical structure of the molecule etc. Generally, a feature selection algorithm is applied before mining, in order to reduce the search space, as well as the model dimension. We do not focus on feature selection methods in this chapter.

The classification approach uses both the topological representation that sees a chemical compound as an undirected graph, having atoms in the vertices and bonds in the edges and the geometric representation that sees a chemical compound as an undirected graph with 3D coordinates attached to the vertices.

The predictive association-based model approach is applied for organic compounds only and uses typical sub-structure presence/count descriptors (a typical substructure can be, for example, - CH_3 or –CH_2-). It also includes a pre-clustered target item, the activity to be predicted.

Resulting Data Model

The frequent sub-structure mining attempts to build, just like frequent itemsets, frequent connected sub-graphs, by adding vertices step-by step, in an Apriori fashion. The main difference from frequent itemset mining is that graph isomorphism has to be checked, in order to correctly compute itemset support in the database. The purpose of frequent sub-structure mining is the classification of chemical compounds, using a Support Vector Machine-based classification algorithm on the chemical compound structure expressed in terms of the resulted frequent sub-structures.

The predictive association rule-based model considers as mining result only the global implications with predictive capability, namely the ones comprising the target item in the rule's conclusion. The prediction is achieved by applying to a new compound all the rules in the model. Some rules may not apply (rule's premises are not satisfied by the compound's structure) and some rules may predict activity clusters. Each cluster can be predicted by a number of rules. After subjecting the compound to the predictive model, it can yield:

- a "none" result, meaning that the compound's activity can not be predicted with the model,
- a cluster id result, meaning the predicted activity cluster,
- several cluster ids; whenever this situation occurs it can be dealt with in various manners: a vote can be held and the majority cluster id can be declared a winner, or the rule set (the model) can be refined since it is too general.

Contribution of Association Rule-Based Approaches for the QSAR Problem

The main advantage in building a classification or prediction model in terms of association rules is model readability. The QSAR problem requires inter-disciplinary team effort, so an important step in validating a resulting model would be to present it to domain experts. A predictive model with no explanatory capabilities can not express the conceptual relationship structure-activity, it can only express a numerical, difficult to understand, relationship.

FUTURE TRENDS

We are considering that the most challenging trends would manifest in:

- extending the above mentioned approaches to other descriptors;
- associating activity behavior with the compound classes resulted from the classification approach;
- building a predictive model on the activity associated-discovered classes.

Both approaches consider, at one stage or another, compounds described by their sub-structures. Association rule mining has been conceived for the market basket analysis; hence it is particularly well fitted to presence data like the sub-structure presence data taken into account by the mentioned techniques. It would be interesting to see if different types of descriptors are suited for these approaches.

The classification approach does not solve the QSAR problem unless activity items are attached to compound classes. The presence of activity items does not guarantee that prediction would be satisfactory within the classes.

The weakest point of predictive model building for the second approach is the pre-clustering of activity items. Building the model using classes previously discovered by the classification approach, may lead to a more reliable prediction technique.

CONCLUSION

We have introduced the idea of descriptive data mining for the QSAR prediction problem in order to add readability to the prediction process. Since the QSAR problem requires domain expertise readability is extremely important. Association rules have the explanatory capability, but they are better suited for presence data, which is not necessarily the case of chemical structure descriptors. The results obtained so far are promising, but QSAR has proven to be a difficult problem, so achieving satisfactory prediction accuracy would have to rely on a profound knowledge of the application domain.

REFERENCES

Agrawal, R., & Srikant, R. (1994, September). Fast algorithms for mining association rules. *Very Large Data Bases 20th International Conference, VLDB'94*, Santiago de Chile, Chile, 487-499.

Agrawal, R., Imielinski, T., & Swami, A. (1993, May). Mining association rules between sets of items in large databases. *Management of Data ACM SIGMOD Conference*, Washington D.C., USA, 207-216.

Deshpande, M., Kuramochi, M., Wale, N. & Karypis, G., (2005) Frequent Substructure-Based Approaches for Classifying Chemical Compounds. *IEEE Transaction on Knowledge and Data Engineering*, 17(8): 1036-1050.

Dumitriu, L., Segal, C., Craciun, M., Cocu, A., & Georgescu, L.P. (2006). Model discovery and validation for the QSAR problem using association rule mining. *Proceedings of ICCS'06*, Volume 11 ISBN:975-00803-0-0, Prague (to appear).

Ganter, B., & Wille, R. (1999). *Formal Concept Analysis –Mathematical Foundations*. Berlin: Springer Verlag.

Kryszkiewicz, M. (1998) *Fast discovery of representative association rules*. Lecture Notes in Artificial Intelligence, volume 1424, pages 214--221. Proceedings of RSCTC 98, Springer-Verlag.

Langdon, W. B. & Barrett, S. J. (2004). Genetic Programming in Data Mining for Drug Discovery. *Evolutionary Computing in Data Mining*, Springer, 2004, Ashish Ghosh and Lakhmi C. Jain, 163, Studies in Fuzziness and Soft Computing, 10, ISBN 3-540-22370-3, pp. 211--235.

Neagu, C.D., Benfenati, E., Gini, G., Mazzatorta, P., Roncaglioni, A., (2002). Neuro-Fuzzy Knowledge Representation for Toxicity Prediction of Organic Compounds. *Proceedings of the 15th European Conference on Artificial Intelligence*, Frank van Harmelen (Ed.):, ECAI'2002, Lyon, France, July 2002. IOS Press 2002: pp. 498-502.

Pasquier, N., Bastide, Y., Taouil, R., & Lakhal, L. (1999, January). Discovering frequent closed itemsets for association rules. *Database Theory International Conference*, ICDT'99, Jerusalem, Israel, 398-416.

Pei, J., Han, J., & Mao, R. (2000, May). CLOSET: An efficient algorithm for mining frequent closed itemsets. *Data Mining and Knowledge Discovery Conference*, DMKD 2000, Dallas, Texas, 11-20.

Wang, Z., Durst, G., Eberhart, R., Boyd, D., & Ben-Miled, Z., (2004). Particle Swarm Optimization and Neural Network Application for QSAR. *Proceedings of the 18th International Parallel and Distributed Processing Symposium (IPDPS 2004)*, 26-30 April 2004, Santa Fe, New Mexico, USA. IEEE Computer Society 2004, ISBN 0-7695-2132-0.

Zaki, M. J., & Ogihara, M. (1998, June). Theoretical foundations of association rules. *In 3rd Research Issues in Data Mining and Knowledge Discovery ACM SIGMOD Workshop*, DMKD'98, Seattle, Washington.

Zaki, M. J., & Hsiao, C. J. (1999). CHARM: An Efficient Algorithm for Closed Association Rule Mining, *Technical Report 99-10*, Department of Computer Science, Rensselaer Polytechnic Institute.

KEY TERMS

Association Rule: Pair of frequent itemsets (A, B), where the ratio between the support of $A \cup B$ and A itemsets is greater than a predefined threshold, denoted *minconf*.

Closure Operator: Let S be a set and c: $\wp(S) \rightarrow \wp(S)$; c is a *closure operator* on S if $\forall$ X, Y $\subseteq$ S, c satisfies the following properties:

1. extension, $X \subseteq c(X)$;
2. mononicity, if $X \subseteq Y$, then $c(X) \subseteq c(Y)$;
3. idempotency, $c(c(X)) = c(X)$.

Note: $s \circ t$ and $t \circ s$ are closure operators, when s and t are the mappings in a Galois connection.

Concept: The Galois connection of the (***T***, ***I***, ***D***) context, a concept is a pair (X, Y), $X \subseteq \boldsymbol{T}$, $Y \subseteq \boldsymbol{I}$, that satisfies s(X)=Y and t(Y)=X. X is called the *extent* and Y the *intent* of the concept (X,Y).

Context: A triple (***T***, ***I***, ***D***) where ***T*** and ***I*** are sets and $\boldsymbol{D} \subseteq \boldsymbol{T} \times \boldsymbol{I}$. The elements of ***T*** are called *objects* and the elements of ***I*** are called *attributes*. For any $t \in \boldsymbol{T}$ and $i \in \boldsymbol{I}$, we note $t\boldsymbol{D}i$ when *t* is related to *i*, *i.e.* (*t*, *i*) $\in \boldsymbol{D}$.

Frequent Itemset: Itemset with support higher than a predefined threshold, denoted *minsup*.

Galois Connection: Let (***T***, ***I***, ***D***) be a context. Then the mappings

s: $\wp(\boldsymbol{T}) \rightarrow \wp(\boldsymbol{I})$, $s(X) = \{ i \in \boldsymbol{I} \mid (\forall t \in X)\ t\boldsymbol{D}i \}$

t: $\wp(\boldsymbol{I}) \rightarrow \wp(\boldsymbol{T})$, $s(Y) = \{ t \in \boldsymbol{T} \mid (\forall i \in Y)\ t\boldsymbol{D}i \}$

define a *Galois connection* between $\wp(\boldsymbol{T})$ and $\wp(\boldsymbol{I})$, the power sets of ***T*** and ***I***, respectively.

Itemset: Set of items in a Boolean database D, $I=\{i_1, i_2, \ldots i_n\}$.

Itemset Support: The ratio between the number of transactions in D comprising all the items in I and the total number of transactions in D ($support(I) = |\{T_i \in D| (\forall i_j \in I)\ i_j \in T_i \}| / |D|$).

Pseudo-Intent: The set X is a pseudo-intent if $X \neq c(X)$, where c is a closure operator, and for all pseudo-intents $Q \subset X$, $c(Q) \subseteq X$.

Association Rule Mining of Relational Data

Anne Denton
North Dakota State University, USA

Christopher Besemann
North Dakota State University, USA

INTRODUCTION

Most data of practical relevance are structured in more complex ways than is assumed in traditional data mining algorithms, which are based on a single table. The concept of relations allows for discussing many data structures such as trees and graphs. Relational data have much generality and are of significant importance, as demonstrated by the ubiquity of relational database management systems. It is, therefore, not surprising that popular data mining techniques, such as association rule mining, have been generalized to relational data. An important aspect of the generalization process is the identification of challenges that are new to the generalized setting.

BACKGROUND

Several areas of databases and data mining contribute to advances in association rule mining of relational data.

- **Relational data model:** Underlies most commercial database technology and also provides a strong mathematical framework for the manipulation of complex data. Relational algebra provides a natural starting point for generalizations of data mining techniques to complex data types.
- **Inductive Logic Programming, ILP (Džeroski & Lavrač, 2001, pp. 48-73):** Treats multiple tables and patterns as logic programs. Hypothesis for generalizing data to unseen examples are solved using first-order logic. Background knowledge is incorporated directly as a program.
- **Association Rule Mining, ARM (Agrawal & Srikant, 1994):** Identifies associations and correlations in large databases. The result of an ARM algorithm is a set of association rules in the form $A \rightarrow C$. There are efficient algorithms such as Apriori that limit the output to sets of items that occur more frequently than a given threshold.
- **Graph Theory:** Addresses networks that consist of nodes that are connected by edges. Traditional graph theoretic problems typically assume no more than one property per node or edge. Solutions to graph-based problems take into account graph and subgraph isomorphism. For example, a subgraph should only count once per isomorphic instance. Data associated with nodes and edges can be modeled within the relational algebra framework.
- **Link-based Mining (Getoor & Diehl, 2005):** Addresses data containing sets of linked objects. The links are exploited in tasks such as object ranking, classification, and link prediction. This work considers multiple relations in order to represent links.

Association rule mining of relational data incorporates important aspects of these areas to form an innovative data mining area of important practical relevance.

MAIN THRUST OF THE CHAPTER

Association rule mining of relational data is a topic that borders on many distinct topics, each with its own opportunities and limitations. Traditional association rule mining allows extracting rules from large data sets without specification of a consequent. Traditional predictive modeling techniques lack this generality and only address a single class label. Association rule mining techniques can be efficient because of the pruning opportunity provided by the downward closure property of support, and through the simple structure of the resulting rules (Agrawal & Srikant, 1994).

When applying association rule mining to relational data, these concepts cannot easily be transferred. This

can be seen particularly easily for data with an underlying graph structure. Graph theory has been developed for the special case of relational data that represent connectivity between nodes or objects with no more than one label. A commonly studied pattern mining problem in graph theory is frequent subgraph discovery (Kuramochi & Karypis, 2004). Challenges in gaining efficiency differ substantially in frequent subgraph discovery compared with data mining of single tables: While downward closure is easy to achieve in single-table data, it requires advanced edge disjoint mining techniques in graph data. On the other hand, while the subgraph isomorphism problem has simple solutions in a graph setting, it cannot easily be discussed in the context of relational joined tables.

This chapter attempts to view the problem of relational association rule mining from the perspective of these and other data mining areas, and highlights challenges and solutions in each case.

General Concept

Two main challenges have to be addressed when applying association rule mining to relational data. Combined mining of multiple tables leads to a search space that is typically large even for moderately sized tables. Performance is, thereby, commonly an important issue in relational data mining algorithms. A less obvious problem lies in the skewing of results (Jensen & Neville, 2007, Getoor & Diehl, 2005). Unlike single-table data, relational data records cannot be assumed to be independent.

One approach to relational data mining is to convert the data from a multiple table format to a single table format using methods such as relational joins and aggregation queries. The relational join operation combines each record from one table with each occurrence of the corresponding record in a second table. That means that the information in one record is represented multiple times in the joined table. Data mining algorithms that operate either explicitly or implicitly on joined tables, thereby, use the same information multiple times. This also applies to algorithms in which tables are joined on-the-fly by identifying corresponding records as they are needed. The relational learning task of transforming multiple relations into propositional or single-table format is also called propositionalization (Kramer et al., 2001). We illustrate specific issues related to reflexive relationships in the next section on relations that represent a graph.

A variety of techniques have been developed for data mining of relational data (Džeroski & Lavrač, 2001). A typical approach is called inductive logic programming, ILP. In this approach relational structure is represented in the form of Prolog queries, leaving maximum flexibility to the user. ILP notation differs from the relational algebra notation; however, all relational operators can be represented in ILP. The approach thereby does not limit the types of problems that can be addressed. It should, however, also be noted that relational database management systems are developed with performance in mind and Prolog-based environments may present limitations in speed.

Application of ARM within the ILP setting corresponds to a search for frequent Prolog (Datalog) queries as a generalization of traditional association rules (Dehaspe & Toivonen, 1999). An example of association rule mining of relational data using ILP (Dehaspe & Toivonen, 2001) could be shopping behavior of customers where relationships between customers are included in the reasoning as in the rule:

{customer(X), parent(X,Y)}→{buys(Y, cola)},

which states that if *X* is a parent then their child *Y* will buy a *cola*. This rule covers tables for the parent, buys, and customer relationships. When a pattern or rule is defined over multiple tables, a relational key is defined as the unit to which queries must be rolled up (usually using the Boolean existential function). In the customer relationships example a key could be "customer", so support is based on the number of customers that support the rule. Summarizations such as this are also needed in link-based classification tasks since individuals are often considered the unknown input examples (Getoor & Diehl, 2005). Propositionalization methods construct features by traversing the relational link structure. Typically, the algorithm specifies how to place the constructed attribute into a single table through the use of aggregation or "roll-up" functions (Kramer et al., 2001). In general, any relationship of a many-to-many type will require the use of aggregation when considering individual objects since an example of a pattern can extend to arbitrarily many examples of a larger pattern. While ILP does not use a relational joining step as such, it does also associate individual objects with multiple occurrences of corresponding

objects. Problems related to skewing are, thereby, also encountered in this approach.

An alternative to the ILP approach is to apply the standard definition of association rule mining to relations that are joined using the relational join operation. While such an approach is less general it is often more efficient since the join operation is highly optimized in standard database systems. It is important to note that a join operation typically changes the support of an item set, and any support calculation should therefore be based on the relation that uses the smallest number of join operations (Cristofor & Simovici, 2001).

Defining rule interest is an important issue in any type of association rule mining. In traditional association rule mining the problem of rule interest has been addressed in a variety of work on redundant rules, including closed set generation (Zaki, 2000). Additional rule metrics such as lift and conviction have been defined (Brin et al., 1997). In relational association rule mining the problem has been approached by the definition of a deviation measure (Dehaspe & Toivonen, 2001). Relational data records have natural dependencies based on the relational link structure. Patterns derived by traversing the link structure will also include dependencies. Therefore it is desirable to develop algorithms that can identify these natural dependencies. Current relational and graph-based pattern mining does not consider intra-pattern dependency. In general it can be noted that relational data mining poses many additional problems related to skewing of data compared with traditional mining on a single table (Jensen & Neville, 2002).

Relations that Represent a Graph

One type of relational data set has traditionally received particular attention, albeit under a different name. A relation representing a relationship between entity instances of the same type, also called a reflexive relationship, can be viewed as the definition of a unipartite graph. Graphs have been used to represent social networks, biological networks, communication networks, and citation graphs, just to name a few. Traditional graph-based approaches focus on connectivity only and are discussed in the related research section.

Recent work extends the field of graph-based patterns to multiple properties on nodes (Oyama et al., 2002; Besemann et al., 2004; Rahal et al., 2006; Besemann et al., 2006; Besemann et al., 2007). Other recent work studies tree based patterns that can represent general graphs by repeating node labels in the tree (Goethals et al., 2005). These graph-based approaches differ from the previous relational approaches in that they do not consider a universal key or record type as the unit of support counts. For example, in (Besemann et al., 2004) the rows for each join definition are considered the transactions therefore the universal key is the join "shape" itself. Relational approaches "roll-up" to a common level such as single nodes. Thus graph-based rule discovery must be performed in a level-by-level basis based on each shape or join operation and by the number of items.

A typical example of an association rule mining problem in graphs is mining of annotation data of proteins in the presence of a protein-protein interaction graph (Oyama et al., 2002). Associations are extracted that relate functions and localizations of one protein with those of interacting proteins. Oyama et al. use association rule mining, as applied to joined relations, for this work. Another example could be association rule mining of attributes associated with scientific publications on the graph of their mutual citations (Rahal et al., 2006).

A problem of the straight-forward approach of mining joined tables directly becomes obvious upon further study of the rules: In most cases the output is dominated by rules that involve the same item as it occurs in different entity instances that participate in a relationship. In the example of protein annotations within the protein interaction graph this is expressed in rules like:

{protein(A), protein(B), interaction(A, B), location(A, nucleus)}→{location(B, nucleus)}

that states if one of two interacting proteins is in the nucleus then the other protein will also be in the nucleus. Similarities among relational neighbors have been observed more generally for relational databases (Macskassy & Provost, 2003). It can be shown that filtering of output is not a consistent solution to this problem, and items that are repeated for multiple nodes should be eliminated in a preprocessing step (Besemann et al., 2004). This is an example of a problem that does not occur in association rule mining of a single table and requires special attention when moving to multiple relations. The example also highlights the

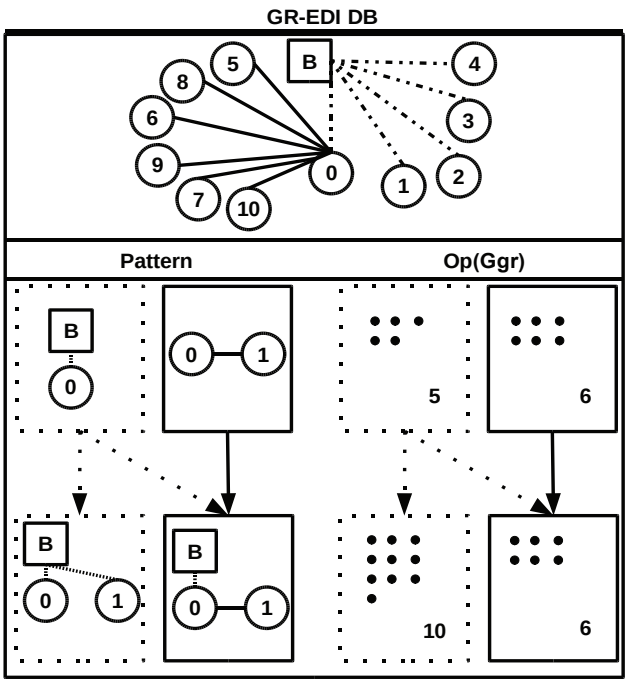
need to discuss what the differences between sets of items of related objects are.

A main problem in applying association rule mining to relational data that represents a graph is the lack of downward closure for graph-subgraph relationships. Edge disjoint instance mining techniques (Kuramochi & Karypis, 2005; Vanetik, 2006; Chen et al, 2007) have been used in frequent subgraph discovery. As a generalization, graphs with sets of labels on nodes have been addressed by considering the node – data relationship as a bipartite graph. Initial work combined the bipartite graph with other graph data as if it were one input graph (Kuramochi & Karypis, 2005). It has been shown in (Besemann & Denton, 2007) that it is beneficial to not include the bipartite graph in the determination of edge disjointness and that downward closure can still be achieved. However, for this to be true, some patterns have to be excluded from the search space. In the following this problem will be discussed in more detail.

Graphs in this context can be described in the form: $G(V, E, L_V, T_V, T_E)$ where the graph vertices $V = \{N \cup D\}$ are composed of entity nodes N and descriptors nodes D. Descriptor nodes correspond to attributes for the entities. The graph edges are $E = \{U \cup B\}$ where $U \subseteq (N \times N)$ is a unipartite relationship between entities and $B \subseteq (N \times D)$ is a bipartite relationship between entities and descriptors. A labeling function L assigns symbols as labels for vertices. Finally T_V and T_E denote the type of vertices and edges as entity or descriptor and unipartite or bipartite respectively. In this context, patterns, which can later be used to build association rules, are simply subgraphs of the same format.

Figure 1 shows a portion of the search space for the GR-EDI algorithm by Besemann and Denton. Potential patterns are arranged in a lattice where child nodes differ from parents by one edge. The left portion describes graph patterns in the space. As mentioned earlier, edge-disjoint instance mining (EDI) approaches allow for downward closure of patterns in the lattice with respect to support. The graph of edge disjoint instances is given for each pattern in the right of the figure. At the level shown, all instances are disjoint therefore the resulting instance graph is composed of individual nodes. This is the case since no pattern contains more than one unipartite (solid) edge and the EDI constraint is only applied to unipartite edges in this case. Dashed boxes indicate patterns that are not guaranteed to meet conditions for the monotone frequency property required for downward closure.

Figure 1. Illustration of need for specifying new pattern constraints when removing edge-disjointness requirement for bipartite edges



As shown, an instance for a pattern with no unipartite edges can be arbitrarily extended to instances of a larger pattern. In order to solve this problem, a pattern constraint must be introduced that requires valid patterns to at least have one unipartite edge connected to each entity node.

Related Research Areas

A related area of research is graph-based pattern mining. Traditional graph-based pattern mining does not produce association rules but rather focuses on the task of frequent subgraph discovery. Most graph-based methods consider a single label or attribute per node. When there are multiple attributes, the data are either modeled with zero or one label per node or as a bipartite graph. One graph-based task addresses multiple graph transactions where the data are a set of graphs (Inokuchi et al, 2000; Yan and Han, 2002; Kuramochi & Karypis, 2004; Hasan et al, 2007). Since each record or transaction is a graph, a subgraph pattern is counted once for each graph in which it exists at least once. In that sense transactional methods are not much different than single-table item set methods.

Single graph settings differ from transactional settings since they contain only one input graph rather than a set of graphs (Kuramochi & Karypis, 2005; Vanetik, 2006; Chen et al, 2007). They cannot use simple existence of a subgraph as the aggregation function; otherwise the pattern supports would be either one or zero. If all examples were counted without aggregation then the problem would no longer satisfy downward closure. Instead, only those instances are counted as discussed in the previous section.

In relational pattern mining multiple items or attributes are associated with each node and the main challenge is to achieve scaling with respect to the number of items per node. Scaling to large subgraphs is usually less relevant due to the "small world" property of many types of graphs. For most networks of practical interest any node can be reached from almost any other by means of no more than some small number of edges (Barabasi & Bonabeau, 2003). Association rules that involve longer distances are therefore unlikely to produce meaningful results.

There are other areas of research on ARM in which related transactions are mined in some combined fashion. Sequential pattern or episode mining (Agrawal & Srikant 1995; Yan, Han, & Afshar, 2003) and inter-transaction mining (Tung et al., 1999) are two main categories. Generally the interest in association rule mining is moving beyond the single-table setting to incorporate the complex requirements of real-world data.

FUTURE TRENDS

The consensus in the data mining community of the importance of relational data mining was recently paraphrased by Dietterich (2003) as "I.i.d. learning is dead. Long live relational learning". The statistics, machine learning, and ultimately data mining communities have invested decades into sound theories based on a single table. It is now time to afford as much rigor to relational data. When taking this step it is important to not only specify generalizations of existing algorithms but to also identify novel questions that may be asked that are specific to the relational setting. It is, furthermore, important to identify challenges that only occur in the relational setting, including skewing due to traversal of the relational link structure and correlations that are frequent in relational neighbors.

CONCLUSION

Association rule mining of relational data is a powerful frequent pattern mining technique that is useful for several data structures including graphs. Two main approaches are distinguished. Inductive logic programming provides a high degree of flexibility, while mining of joined relations is a fast technique that allows the study of problems related to skewed or uninteresting results. The potential computational complexity of relational algorithms and specific properties of relational data make its mining an important current research topic. Association rule mining takes a special role in this process, being one of the most important frequent pattern algorithms.

REFERENCES

Agrawal, R. & Srikant, R. (1994). Fast Algorithms for Mining Association Rules in Large Databases. In *Proceedings of the 20th international Conference on Very Large Data Bases*, San Francisco, CA, 487-499.

Agrawal, R. & Srikant, R. (1995). Mining sequential patterns. In Proceedings *11th International Conference on Data Engineering*, IEEE Computer Society Press, Taipei, Taiwan, 3-14.

Barabasi, A.L. & Bonabeau, E. (2003). Scale-free Networks, *Scientific American*, 288(5), 60-69.

Besemann, C. & Denton, A. (Apr. 2007). Mining edge-disjoint patterns in graph-relational data. *Workshop on Data Mining for Biomedical Informatics in conjunction with the 6th SIAM International Conference on Data Mining*, Minneapolis, MN.

Besemann, C., Denton, A., Carr, N.J., & Prüß, B.M. (2006). BISON: A Bio- Interface for the Semi-global analysis Of Network patterns, *Source Code for Biology and Medicine*, 1:8.

Besemann, C., Denton, A., Yekkirala, A., Hutchison, R., & Anderson, M. (Aug. 2004). Differential Association Rule Mining for the Study of Protein-Protein Interaction Networks. In Proceedings *ACM SIGKDD Workshop on Data Mining in Bioinformatics*, Seattle, WA.

Brin, S., Motwani, R., Ullman, J.D., & Tsur, S. (1997). Dynamic itemset counting and implication rules for market basket data. In Proceedings *ACM SIGMOD International Conference on Management of Data*, Tucson, AZ.

Chen, C., Yan, X., Zhu, F., & Han, J. (2007). gApprox: Mining Frequent Approximate Patterns from a Massive Network. In Proceedings *International Conference on Data Mining*, Omaha, NE.

Cristofor, L. & Simovici, D. (2001). Mining Association Rules in Entity-Relationship Modeled Databases, Technical Report, University of Massachusetts Boston.

Dehaspe, L. & De Raedt, L. (Dec. 1997). Mining Association Rules in Multiple Relations. In Proceedings *7th International Workshop on Inductive Logic Programming*", Prague, Czech Republic", 125-132.

Dehaspe, L. & Toivonen, H. (1999). Discovery of frequent DATALOG patterns. Data Mining and Knowledge Discovery 3(1).

Dehaspe, L. & Toivonen, H. (2001). Discovery of Relational Association Rules. In *Relational Data Mining*. Eds. Džeroski, S. & Lavrač, N. Berlin: Springer.

Dietterich, T. (2003). Sequential Supervised Learning: Methods for Sequence Labeling and Segmentation. Invited Talk, *3rd IEEE International Conference on Data Mining*, Melbourne, FL, USA.

Džeroski, S. & Lavrač, N. (2001). *Relational Data Mining*, Berlin: Springer.

Getoor, L. & Diehl, C. (2005). Link mining: a survey. SIGKDD Explorer Newsletter 7(2) 3-12.

Goethals, B., Hoekx, E., & Van den Bussche, J. (2005). Mining tree queries in a graph. In Proceeding 11th International Conference on Knowledge Discovery in Data Mining, Chicago, Illinois, USA. 61-69.

Hasan, M., Chaoji, V., Salem, S., Besson, J., & Zaki, M.J. (2007). ORIGAMI: Mining Representative Orthogonal Graph Patterns. In Proceedings *International Conference on Data Mining*, Omaha, NE.

Inokuchi, A., Washio, T. & Motoda, H. (2000). An Apriori-Based Algorithm for Mining Frequent Substructures from Graph Data. In Proceedings *4th European Conference on Principles of Data Mining and Knowledge Discovery*. Lyon, France", 13-23.

Jensen, D. & Neville, J. (2002). Linkage and autocorrelation cause feature selection bias in relational learning. In Proceedings *19th International Conference on Machine Learning*, Sydney, Australia, 259-266.

Kramer, S., Lavrač, N. & Flach, P. Propositionalization Approaches to Relational Data Mining. In *Relational Data Mining*. Eds. Džeroski, S. & Lavrač, N. Berlin: Springer.

Kuramochi, M. & Karypis, G. (2004). An Efficient Algorithm for Discovering Frequent Subgraphs. IEEE Transactions on Knowledge and Data Engineering. **16**(9).

Kuramochi, M. and Karypis, G. (2005). Finding Frequent Patterns in a Large Sparse Graph, *Data Mining and Knowledge Discovery.* 11(3).

Macskassy, S. & Provost, F. (2003). A Simple Relational Classifier. In Proceedings *2nd Workshop on Multi-Relational Data Mining at KDD'03*, Washington, D.C.

Neville, J. and Jensen, D. (2007). Relational Dependency Networks. Journal of Machine Learning Research. **8**(Mar) 653-692.

Oyama, T., Kitano, K., Satou, K. & Ito,T. (2002). Extraction of knowledge on protein-protein interaction by association rule discovery. Bioinformatics **18**(8) 705-714.

Rahal, I., Ren, D., Wu, W., Denton, A., Besemann, C., & Perrizo, W. (2006). Exploiting edge semantics in citation graphs using efficient, vertical ARM. Knowledge and Information Systems. **10**(1).

Tung, A.K.H., Lu, H. Han, J., & Feng, L. (1999). Breaking the Barrier of Transactions: Mining Inter-Transaction Association Rules. In Proceedings *International Conference on Knowledge Discovery and Data Mining*, San Diego, CA.

Vanetik, N., Shimony, S. E., & Gudes, E. (2006). Support measures for graph data. Data Mining and Knowledge Discovery 13(2) 243-260.

Yan, X. & Han, J. (2002). gSpan: Graph-based substructure pattern mining. In Proceedings *International Conference on Data Mining*, Maebashi City, Japan.

Yan, X., Han, J., & Afshar, R. (2003). CloSpan: Mining Closed Sequential Patterns in Large Datasets. In Proceedings *2003 SIAM International Conference on Data Mining*, San Francisco, CA.

Zaki, M.J. (2000). Generating non-redundant association rules. In Proceedings *International Conference on Knowledge Discovery and Data Mining*, Boston, MA, 34-43.

KEY TERMS

Antecedent: The set of items A in the association rule $A \rightarrow C$.

Apriori: Association rule mining algorithm that uses the fact that the support of a non-empty subset of an item set cannot be smaller than the support of the item set itself.

Association Rule: A rule of the form $A \rightarrow C$ meaning "if the set of items A is present in a transaction, then the set of items C is likely to be present too". A typical example constitutes associations between items purchased at a supermarket.

Confidence: The confidence of a rule $A \rightarrow C$ is support($A \cup C$) / support(A) that can be viewed as the sample probability Pr($C|A$).

Consequent: The set of items C in the association rule $A \rightarrow C$.

Entity-Relationship Model (E-R-Model): A model to represent real-world requirements through entities, their attributes, and a variety of relationships between them. E-R-Models can be mapped automatically to the relational model.

Inductive Logic Programming (ILP): Research area at the interface of machine learning and logic programming. Predicate descriptions are derived from examples and background knowledge. All examples, background knowledge and final descriptions are represented as logic programs.

Redundant Association Rule: An association rule is redundant if it can be explained based entirely on one or more other rules.

Relational Database: A database that has relations and relational algebra operations as underlying mathematical concepts. All relational algebra operations result in relations as output. A join operation is used to combine relations. The concept of a relational database was introduced by E. F. Codd at IBM in 1970.

Relation: A mathematical structure similar to a table in which every row is unique, and neither rows nor columns have a meaningful order.

Support: The support of an item set is the fraction of transactions or records that have all items in that item set. Absolute support measures the count of transactions that have all items.

Association Rules and Statistics

Martine Cadot
University of Henri Poincaré/LORIA, Nancy, France

Jean-Baptiste Maj
LORIA/INRIA, France

Tarek Ziadé
NUXEO, France

INTRODUCTION

A manager would like to have a dashboard of his company without manipulating data. Usually, statistics have solved this challenge, but nowadays, data have changed (Jensen, 1992); their size has increased, and they are badly structured (Han & Kamber, 2001). A recent method—data mining—has been developed to analyze this type of data (Piatetski-Shapiro, 2000). A specific method of data mining, which fits the goal of the manager, is the extraction of association rules (Hand, Mannila & Smyth, 2001). This extraction is a part of attribute-oriented induction (Guyon & Elisseeff, 2003).

The aim of this paper is to compare both types of extracted knowledge: association rules and results of statistics.

BACKGROUND

Statistics have been used by people who want to extract knowledge from data for one century (Freeman, 1997). Statistics can describe, summarize and represent the data. In this paper data are structured in tables, where lines are called objects, subjects or transactions and columns are called variables, properties or attributes. For a specific variable, the value of an object can have different types: quantitative, ordinal, qualitative or binary. Furthermore, statistics tell if an effect is significant or not. They are called inferential statistics.

Data mining (Srikant, 2001) has been developed to precede a huge amount of data, which is the result of progress in digital data acquisition, storage technology, and computational power. The association rules, which are produced by data-mining methods, express links on database attributes. The knowledge brought by the association rules is shared in two different parts. The first describes general links, and the second finds specific links (knowledge nuggets) (Fabris & Freitas, 1999; Padmanabhan & Tuzhilin, 2000). In this article, only the first part is discussed and compared to statistics. Furthermore, in this article, only data structured in tables are used for association rules.

MAIN THRUST

The problem differs with the number of variables. In the sequel, problems with two, three, or more variables are discussed.

Two Variables

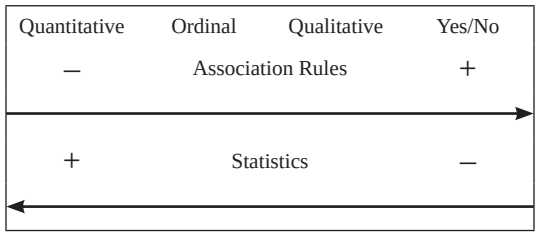
The link between two variables (A and B) depends on the coding. The outcome of statistics is better when data are quantitative. A current model is linear regression. For instance, the salary (S) of a worker can be expressed by the following equation:

$$S = 100\ Y + 20000 + \varepsilon \quad (1)$$

where Y is the number of years in the company, and ε is a random number. This model means that the salary of a newcomer in the company is $20,000 and increases by $100 per year.

The association rule for this model is: $Y \rightarrow S$. This means that there are a few senior workers with a small paycheck. For this, the variables are translated into binary variables. Y is not the number of years, but the property has seniority, which is not quantitative but of type Yes/No. The same transformation is applied to the salary S, which becomes the property "has a big salary."

Figure 1. Coding and analysis methods



Therefore, these two methods both provide the link between the two variables and have their own instruments for measuring the quality of the link. For statistics, there are the tests of regression model (Baillargeon, 1996), and for association rules, there are measures like support, confidence, and so forth (Kodratoff, 2001). But, depending on the type of data, one model is more appropriate than the other (Figure 1).

Three Variables

If a third variable E, the experience of the worker, is integrated, the equation (1) becomes:

$$S = 100\ Y + 2000\ E + 19000 + \varepsilon \qquad (2)$$

E is the property "has experience." If E=1, a new experienced worker gets a salary of $21,000, and if E=0, a new non-experienced worker gets a salary of $19,000. The increase of the salary, as a function of seniority (Y), is the same in both cases of experience.

$$S = 50\ Y + 1500\ E + 50\ E \acute{}\ Y + 19500 + \varepsilon \qquad (3)$$

Now, if E=1, a new experienced worker gets a salary of $21,000, and if E=0, a new non-experienced worker gets a salary of $19,500. The increase of the salary, as a function of seniority (Y), is $50 higher for experienced workers. These regression models belong to a linear model of statistics (Prum, 1996), where, in the equation (3), the third variable has a particular effect on the link between Y and S, called *interaction* (Winer, Brown & Michels, 1991).

The association rules for this model are:

- $Y \rightarrow S$, $E \rightarrow S$ for the equation (2)
- $Y \rightarrow S$, $E \rightarrow S$, $YE \rightarrow S$ for the equation (3)

The statistical test of the regression model allows to choose with or without interaction (2) or (3). For the association rules, it is necessary to prune the set of three rules, because their measures do not give the choice between a model of two rules and a model of three rules (Zaki, 2000; Zhu, 1998).

More Variables

With more variables, it is difficult to use statistical models to test the link between variables (Megiddo & Srikant, 1998). However, there are still some ways to group variables: clustering, factor analysis, and taxonomy (Govaert, 2003). But the complex links between variables, like interactions, are not given by these models and decrease the quality of the results.

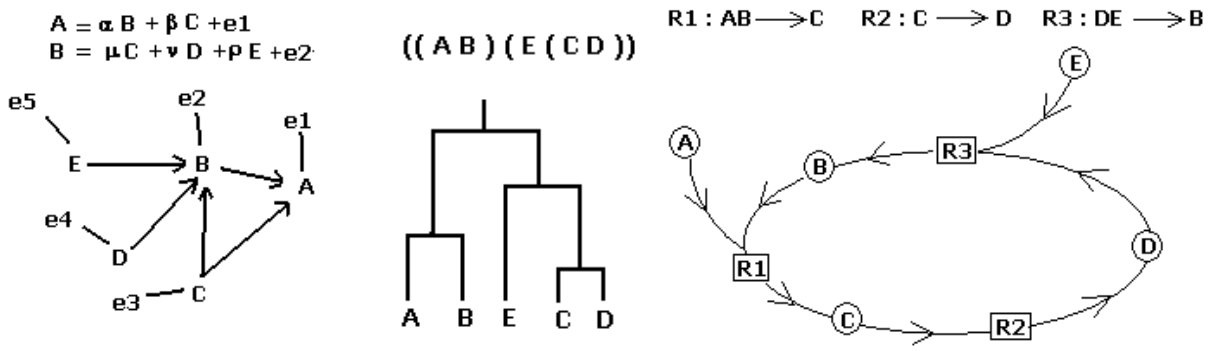
Comparison

Table 1 briefly compares statistics with the association rules. Two types of statistics are described: by tests and by taxonomy. Statistical tests are applied to a small amount of variables and the taxonomy to a great

Table 1. Comparison between statistics and association rules

	Statistics		Data mining
	Tests	**Taxonomy**	**Association rules**
Decision	**Tests (+)**	**Threshold defined (-)**	**Threshold defined (-)**
Level of Knowledge	**Low (-)**	**High and simple (+)**	**High and complex (+)**
Nb. of Variables	**Small (-)**	**High (+)**	**Small and high (+)**
Complex Link	**Yes (-)**	**No (+)**	**No (-)**

Figure 2. a) Regression equations b) Taxonomy c) Association rules



(a) (b) (c)

amount of variables. In statistics, the decision is easy to make out of test results, unlike association rules, where a difficult choice on several indices thresholds has to be performed. For the level of knowledge, the statistical results need more interpretation relative to the taxonomy and the association rules.

Finally, graphs of the regression equations (Hayduk, 1987), taxonomy (Foucart, 1997), and association rules (Gras & Bailleul, 2001) are depicted in Figure 2.

FUTURE TRENDS

With association rules, some researchers try to find the right indices and thresholds with stochastic methods. More development needs to be done in this area. Another sensitive problem is the set of association rules that is not made for deductive reasoning. One of the most common solutions is the pruning to suppress redundancies, contradictions and loss of transitivity. Pruning is a new method and needs to be developed.

CONCLUSION

With association rules, the manager can have a fully detailed dashboard of his or her company without manipulating data. The advantage of the set of association rules relative to statistics is a high level of knowledge. This means that the manager does not have the inconvenience of reading tables of numbers and making interpretations. Furthermore, the manager can find knowledge nuggets that are not present in statistics.

The association rules have some inconvenience; however, it is a new method that still needs to be developed.

REFERENCES

Baillargeon, G. (1996). *Méthodes statistiques de l'ingénieur: Vol. 2*. Trois-Riveres, Quebec: Editions SMG.

Fabris,C., & Freitas, A. (1999). Discovery surprising patterns by detecting occurrences of Simpson's paradox: Research and development in intelligent systems XVI. *Proceedings of the 19th Conference of Knowledge-Based Systems and Applied Artificial Intelligence*, Cambridge, UK.

Foucart, T. (1997). *L'analyse des données, mode d'emploi*. Rennes, France: Presses Universitaires de Rennes.

Freedman, D. (1997). *Statistics*. W.W. New York: Norton & Company.

Govaert, G. (2003). *Analyse de données*. Lavoisier, France: Hermes-Science.

Gras, R., & Bailleul, M. (2001). La fouille dans les données par la méthode d'analyse statistique implicative. *Colloque de Caen*. Ecole polytechnique de l'Université de Nantes, Nantes, France.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection: Special issue on variable and feature selection. *Journal of Machine Learning Research, 3*, 1157-1182.

Han, J., & Kamber, M. (2001). *Data mining: Concepts and techniques*. San Francisco, CA: Morgan Kaufmann.

Hand, D., Mannila, H., & Smyth, P. (2001). *Principles of data mining*. Cambridge, MA: MIT Press.

Hayduk, L.A. (1987). *Structural equation modelling with LISREL*. Maryland: John Hopkins Press.

Jensen, D. (1992). *Induction with randomization testing: Decision-oriented analysis of large data sets* [doctoral thesis]. Washington University, Saint Louis, MO.

Kodratoff, Y. (2001). Rating the interest of rules induced from data and within texts. *Proceedings of the 12th IEEE International Conference on Database and Expert Systems Aplications-Dexa*, Munich, Germany.

Megiddo, N., & Srikant, R. (1998). Discovering predictive association rules. *Proceedings of the Conference on Knowledge Discovery in Data*, New York.

Padmanabhan, B., & Tuzhilin, A. (2000). Small is beautiful: Discovering the minimal set of unexpected patterns. *Proceedings of the Conference on Knowledge Discovery in Data*. Boston, Massachusetts.

Piatetski-Shapiro, G. (2000). Knowledge discovery in databases: 10 years after. *Proceedings of the Conference on Knowledge Discovery in Data*, Boston, Massachusetts.

Prum, B. (1996). *Modèle linéaire: Comparaison de groupes et régression*. Paris, France: INSERM.

Srikant, R. (2001). Association rules: Past, present, future. *Proceedings of the Workshop on Concept Lattice-Based Theory, Methods and Tools for Knowledge Discovery in Databases,* California.

Winer, B.J., Brown, D.R., & Michels, K.M.(1991). *Statistical principles in experimental design*. New York: McGraw-Hill.

Zaki, M.J. (2000). Generating non-redundant association rules. *Proceedings of the Conference on Knowledge Discovery in Data*, Boston, Massachusetts.

Zhu, H. (1998). *On-line analytical mining of association rules* [doctoral thesis]. Simon Fraser University, Burnaby, Canada.

KEY TERMS

Attribute-Oriented Induction: Association rules, classification rules, and characterization rules are written with attributes (i.e., variables). These rules are obtained from data by induction and not from theory by deduction.

Badly Structured Data: Data, like texts of corpus or log sessions, often do not contain explicit variables. To extract association rules, it is necessary to create variables (e.g., keyword) after defining their values (frequency of apparition in corpus texts or simply apparition/non apparition).

Interaction: Two variables, A and B, are in interaction if their actions are not seperate.

Linear Model: A variable is fitted by a linear combination of other variables and interactions between them.

Pruning: The algorithms of extraction for the association rule are optimized in computationality cost but not in other constraints. This is why a suppression has to be performed on the results that do not satisfy special constraints.

Structural Equations: System of several regression equations with numerous possibilities. For instance, a same variable can be made into different equations, and a latent (not defined in data) variable can be accepted.

Taxonomy: This belongs to clustering methods and is usually represented by a tree. Often used in life categorization.

Tests of Regression Model: Regression models and analysis of variance models have numerous hypothesis, e.g. normal distribution of errors. These constraints allow to determine if a coefficient of regression equation can be considered as null with a fixed level of significance.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 74-77, copyright 2005 by IGI Publishing, formerly known as Idea Group Publishing (an imprint of IGI Global).

Audio and Speech Processing for Data Mining

Zheng-Hua Tan
Aalborg University, Denmark

INTRODUCTION

The explosive increase in computing power, network bandwidth and storage capacity has largely facilitated the production, transmission and storage of multimedia data. Compared to alpha-numeric database, non-text media such as audio, image and video are different in that they are unstructured by nature, and although containing rich information, they are not quite as expressive from the viewpoint of a contemporary computer. As a consequence, an overwhelming amount of data is created and then left unstructured and inaccessible, boosting the desire for efficient content management of these data. This has become a driving force of multimedia research and development, and has lead to a new field termed multimedia data mining. While text mining is relatively mature, mining information from non-text media is still in its infancy, but holds much promise for the future.

In general, data mining the process of applying analytical approaches to large data sets to discover implicit, previously unknown, and potentially useful information. This process often involves three steps: data preprocessing, data mining and postprocessing (Tan, Steinbach, & Kumar, 2005). The first step is to transform the raw data into a more suitable format for subsequent data mining. The second step conducts the actual mining while the last one is implemented to validate and interpret the mining results.

Data preprocessing is a broad area and is the part in data mining where essential techniques are highly dependent on data types. Different from textual data, which is typically based on a written language, image, video and some audio are inherently non-linguistic. Speech as a spoken language lies in between and often provides valuable information about the subjects, topics and concepts of multimedia content (Lee & Chen, 2005). The language nature of speech makes information extraction from speech less complicated yet more precise and accurate than from image and video. This fact motivates content based speech analysis for multimedia data mining and retrieval where audio and speech processing is a key, enabling technology (Ohtsuki, Bessho, Matsuo, Matsunaga, & Kayashi, 2006). Progress in this area can impact numerous business and government applications (Gilbert, Moore, & Zweig, 2005). Examples are discovering patterns and generating alarms for intelligence organizations as well as for call centers, analyzing customer preferences, and searching through vast audio warehouses.

BACKGROUND

With the enormous, ever-increasing amount of audio data (including speech), the challenge now and in the future becomes the exploration of new methods for accessing and mining these data. Due to the non-structured nature of audio, audio files must be annotated with structured metadata to facilitate the practice of data mining. Although manually labeled metadata to some extent assist in such activities as categorizing audio files, they are insufficient on their own when it comes to more sophisticated applications like data mining. Manual transcription is also expensive and in many cases outright impossible. Consequently, automatic metadata generation relying on advanced processing technologies is required so that more thorough annotation and transcription can be provided. Technologies for this purpose include audio diarization and automatic speech recognition. Audio diarization aims at annotating audio data through segmentation, classification and clustering while speech recognition is deployed to transcribe speech. In addition to these is event detection, such as, for example, applause detection in sports recordings. After audio is transformed into various symbolic streams, data mining techniques can be applied to the streams to find patterns and associations, and information retrieval techniques can be applied for the purposes of indexing, search and retrieval. The procedure is analogous to video data mining and retrieval (Zhu, Wu, Elmagarmid, Feng, & Wu, 2005; Oh, Lee, & Hwang, 2005).

Diarization is the necessary, first stage in recognizing speech mingled with other audios and is an important field in its own right. The state-of-the-art system has achieved a speaker diarization error of less than 7% for broadcast news shows (Tranter & Reynolds, 2006).

A recent, notable research project on speech transcription is the Effective Affordable Reusable Speech-To-Text (EARS) program (Chen, Kingsbury, Mangu, Povey, Saon, Soltau, & Zweig, 2006). The EARS program focuses on automatically transcribing natural, unconstrained human-human speech from broadcasts and telephone conversations in multiple languages. The primary goal is to generate rich and accurate transcription both to enable computers to better detect, extract, summarize, and translate important information embedded in the speech and to enable humans to understand the speech content by reading transcripts instead of listening to audio signals. To date, accuracies for broadcast news and conversational telephone speech are approximately 90% and 85%, respectively. For reading or dictated speech, recognition accuracy is much higher, and depending on several configurations, it can reach as high as 99% for large vocabulary tasks.

Progress in audio classification and categorization is also appealing. In a task of classifying 198 sounds into 16 classes, (Lin, Chen, Truong, & Chang, 2005) achieved an accuracy of 97% and the performance was 100% when considering Top 2 matches. The 16 sound classes are alto-trombone, animals, bells, cello-bowed, crowds, female, laughter, machines, male, oboe, percussion, telephone, tubular-bells, violin-bowed, violin-pizz and water.

The technologies at this level are highly attractive for many speech data mining applications. The question we ask here is what is speech data mining? The fact is that we have areas close to or even overlapping with it, such as spoken document retrieval for search and retrieval (Hansen, Huang, Zhou, Seadle, Deller, Gurijala, Kurimo, & Angkititrakul, 2005). At this early stage of research, the community does not show a clear intention to segregate them, though. The same has happened with text data mining (Hearst, 1999). In this chapter we define speech data mining as the nontrivial extraction of hidden and useful information from masses of speech data. The same applies to audio data mining. Interesting information includes trends, anomalies and associations with the purpose being primarily for decision making. An example is mining spoken dialog to generate alerts.

MAIN FOCUS

In this section we discuss some key topics within or related to speech data mining. We cover audio diarization, robust speech recognition, speech data mining and spoken document retrieval. Spoken document retrieval is accounted for since the subject is so closely related to speech data mining, and the two draw on each other by sharing many common preprocessing techniques.

Audio Diarization

Audio diarization aims to automatically segment an audio recording into homogeneous regions. Diarization first segments and categorizes audio as speech and non-speech. Non-speech is a general category covering silence, music, background noise, channel conditions and so on. Speech segments are further annotated through speaker diarization which is the current focus in audio diarization. Speaker diarization, also known as "Who Spoke When" or speaker segmentation and clustering, partitions speech stream into uniform segments according to speaker identity.

A typical diarization system comprises such components as speech activity detection, change detection, gender and bandwidth identification, speaker segmentation, speaker clustering, and iterative re-segmentation or boundary refinement (Tranter & Reynolds, 2006). Two notable techniques applied in this area are Gaussian mixture model (GMM) and Bayesian information criterion (BIC), both of which are deployed through the process of diarization. The performance of speaker diarization is often measured by diarization error rate which is the sum of speaker error, missed speaker and false alarm speaker rates.

Diarization is an important step for further processing such as audio classification (Lu, Zhang, & Li, 2003), audio clustering (Sundaram & Narayanan, 2007), and speech recognition.

Robust Speech Recognition

Speech recognition is the process of converting a speech signal to a word sequence. Modern speech recognition systems are firmly based on the principles of statistical pattern recognition, in particular the use of hidden Markov models (HMMs). The objective is to find the most likely sequence of words $\hat{W}$, given the observation data Y which are feature vectors extracted

from an utterance. It is achieved through the following Bayesian decision rule:

$$\hat{W} = \arg\max_{W} P(W \mid Y) = \arg\max_{W} P(W)P(Y \mid W)$$

where $P(W)$ is the a priori probability of observing some specified word sequence W and is given by a language model, for example tri-grams, and $P(Y|W)$ is the probability of observing speech data Y given word sequence W and is determined by an acoustic model, often being HMMs.

HMM models are trained on a collection of acoustic data to characterize the distributions of selected speech units. The distributions estimated on training data, however, may not represent those in test data. Variations such as background noise will introduce mismatches between training and test conditions, leading to severe performance degradation (Gong, 1995). Robustness strategies are therefore demanded to reduce the mismatches. This is a significant challenge placed by various recording conditions, speaker variations and dialect divergences. The challenge is even more significant in the context of speech data mining, where speech is often recorded under less control and has more unpredictable variations. Here we put an emphasis on robustness against noise.

Noise robustness can be improved through feature-based or model-based compensation or the combination of the two. Feature compensation is achieved through three means: feature enhancement, distribution normalization and noise robust feature extraction. Feature enhancement attempts to clean noise-corrupted features, as in spectral subtraction. Distribution normalization reduces the distribution mismatches between training and test speech; cepstral mean subtraction and variance normalization are good examples. Noise robust feature extraction includes improved mel-frequency cepstral coefficients and completely new features. Two classes of model domain methods are model adaptation and multi-condition training (Xu, Tan, Dalsgaard, & Lindberg, 2006).

Speech enhancement unavoidably brings in uncertainties and these uncertainties can be exploited in the HMM decoding process to improve its performance. Uncertain decoding is such an approach in which the uncertainty of features introduced by the background noise is incorporated in the decoding process by using a modified Bayesian decision rule (Liao & Gales, 2005). This is an elegant compromise between feature-based and model-based compensation and is considered an interesting addition to the category of joint feature and model domain compensation which contains well-known techniques such as missing data and weighted Viterbi decoding.

Another recent research focus is on robustness against transmission errors and packet losses for speech recognition over communication networks (Tan, Dalsgaard, & Lindberg, 2007). This becomes important when there are more and more speech traffic through networks.

Speech Data Mining

Speech data mining relies on audio diarization, speech recognition and event detection for generating data description and then applies machine learning techniques to find patterns, trends, and associations.

The simplest way is to use text mining tools on speech transcription. Different from written text, however, textual transcription of speech is inevitably erroneous and lacks formatting such as punctuation marks. Speech, in particular spontaneous speech, furthermore contains hesitations, repairs, repetitions, and partial words. On the other hand, speech is an information-rich media including such information as language, text, meaning, speaker identity and emotion. This characteristic lends a high potential for data mining to speech, and techniques for extracting various types of information embedded in speech have undergone substantial development in recent years.

Data mining can be applied to various aspects of speech. As an example, large-scale spoken dialog systems receive millions of calls every year and generate terabytes of log and audio data. Dialog mining has been successfully applied to these data to generate alerts (Douglas, Agarwal, Alonso, Bell, Gilbert, Swayne, & Volinsky, 2005). This is done by labeling calls based on subsequent outcomes, extracting features from dialog and speech, and then finding patterns. Other interesting work includes semantic data mining of speech utterances and data mining for recognition error detection.

Whether speech summarization is also considered under this umbrella is a matter of debate, but it is nevertheless worthwhile to refer to it here. Speech summarization is the generation of short text summaries of speech (Koumpis & Renals, 2005). An intuitive approach is to apply text-based methods to speech

transcription while more sophisticated approaches combine prosodic, acoustic and language information with textual transcription.

Spoken Document Retrieval

Spoken document retrieval is turned into a text information retrieval task by using a large-vocabulary continuous speech recognition (LVCSR) system to generate a textual transcription. This approach has shown good performance for high-quality, close-domain corpora, for example broadcast news, where a moderate word error rate can be achieved. When word error rate is below one quarter, spoken document retrieval systems are able to get retrieval accuracy similar to using human reference transcriptions because query terms are often long words that are easy to recognize and are often repeated several times in the spoken documents. However, the LVCSR approach presents two inherent deficiencies: vocabulary limitations and recognition errors. First, this type of system can only process words within the predefined vocabulary of the recognizer. If any out-of-vocabulary words appear in the spoken documents or in the queries, the system cannot deal with them. Secondly, speech recognition is error-prone and has an error rate of typically ranging from five percent for clean, close-domain speech to as much as 50 percent for spontaneous, noisy speech, and any errors made in the recognition phase cannot be recovered later on as speech recognition is an irreversible process. To resolve the problem of high word error rates, recognition lattices can be used as an alternative for indexing and search (Chelba, Silva, & Acero, 2007). The key issue in this field is indexing, or more generally, the combination of automatic speech recognition and information retrieval technologies for optimum overall performance.

Search engines have recently experienced a great success in searching and retrieving text documents. Nevertheless the World Wide Web is very silent with only primitive audio search mostly relying on surrounding text and editorial metadata. Content based search for audio material, more specifically spoken archives, is still an uncommon practice and the gap between audio search and text search performance is significant. Spoken document retrieval is identified as one of the key elements of next-generation search engines.

FUTURE TRENDS

The success of speech data mining highly depends on the progress of audio and speech processing. While the techniques have already shown good potentials for data mining applications, further advances are called for. To be applied to gigantic data collected under diverse conditions, faster and more robust speech recognition systems are required.

At present hidden Markov model is the dominating approach for automatic speech recognition. A recent trend worth noting is the revisit of template-based speech recognition, which is an aged, almost obsolete paradigm. It has now been put into a different perspective and framework and is gathering momentum. In addition to this are knowledge based approaches and cognitive science oriented approaches.

In connection with diarization, it is foreseen that there should be a wider scope of studies to cover larger and broader databases, and to gather more diversity of information including emotion, speaker identities and characteristics.

Data mining is often used for surveillance applications, where real-time or near real-time operation is mandatory while at present many systems and approaches work in a batch mode or far from real-time.

Another topic under active research is music data mining for detecting melodic or harmonic patterns from music data. Main focuses are on feature extraction, similarity measures, categorization and clustering, pattern extraction and knowledge representation.

Multimedia data in general contains several different types of media. Information fusion from multiple media has been less investigated and should be given more attention in the future.

CONCLUSION

This chapter reviews audio and speech processing technologies for data mining with an emphasis on speech data mining. The status and the challenges of various related techniques are discussed. The underpinning techniques for mining audio and speech are audio diarization, robust speech recognition, and audio classification and categorization. These techniques have reached a level where highly attractive data

mining applications can be deployed for the purposes of prediction and knowledge discovery. The field of audio and speech data mining is still in its infancy, but it has received attention in security, commercial and academic fields.

REFERENCES

Chelba, C., Silva, J., & Acero, A. (2007). Soft indexing of speech content for search in spoken documents. *Computer Speech and Language*, 21(3), 458-478.

Chen, S.F., Kingsbury, B., Mangu, L., Povey, D., Saon, G., Soltau, H., & Zweig, G. (2006). Advances in speech transcription at IBM under the DARPA EARS program. *IEEE Transactions on Audio, Speech and Language Processing*, 14(5), 1596- 1608.

Douglas, S., Agarwal, D., Alonso, T., Bell, R., Gilbert, M., Swayne, D.F., & Volinsky, C. (2005). Mining customer care dialogs for "daily news". *IEEE Transactions on Speech and Audio Processing*, 13(5), 652-660.

Gilbert, M., Moore, R.K., & Zweig, G. (2005). Editorial – introduction to the special issue on data mining of speech, audio, and dialog. *IEEE Transactions on Speech and Audio Processing*, 13(5), 633-634.

Gong, Y. (1995). Speech recognition in noisy environments: a survey. *Speech Communication*, 16(3), 261-291.

Hansen, J.H.L., Huang, R., Zhou, B., Seadle, M., Deller, J.R., Gurijala, A.R., Kurimo, M., & Angkititrakul, P. (2005). SpeechFind: advances in spoken document retrieval for a national gallery of the spoken word. *IEEE Transactions on Speech and Audio Processing*, 13(5), 712-730.

Hearst, M.A. (1999). Untangling text data mining. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, 3– 10.

Koumpis, K., & Renals, S., (2005). Automatic summarization of voicemail messages using lexical and prosodic features. *ACM Transactions on Speech and Language Processing*, 2(1), 1-24.

Lee, L.-S., & Chen, B. (2005). Spoken document understanding and organization. *IEEE Signal Processing Magazine*, 22(5), 42-60.

Liao, H., & Gales, M.J.F. (2005). Joint uncertainty decoding for noise robust speech recognition. In *Proceedings of INTERSPEECH 2005*, 3129–3132.

Lin, C.-C., Chen, S.-H., Truong, T.-K., & Chang, Y. (2005). Audio classification and categorization based on wavelets and support vector machine. *IEEE Transactions on Speech and Audio Processing*, 13(5), 644-651.

Lu, L., Zhang, H.-J., & Li, S. (2003). Content-based audio classification and segmentation by using support vector machines. *ACM Multimedia Systems Journal* 8(6), 482-492.

Oh, J.H., Lee, J., & Hwang, S. (2005). Video data mining. In J. Wang (Ed.), *Encyclopedia of Data Warehousing and Mining*. Idea Group Inc. and IRM Press.

Ohtsuki, K., Bessho, K., Matsuo, Y., Matsunaga, S., & Kayashi, Y. (2006). Automatic multimedia indexing. *IEEE Signal Processing Magazine*, 23(2), 69-78.

Sundaram, S., & Narayanan, S. (2007). Analysis of audio clustering using word descriptions. In *Proceedings of the 32nd International Conference on Acoustics, Speech, and Signal Processing*.

Tan, P.-N., Steinbach, M., & Kumar, V. (2005). *Introduction to data mining*. Addison-Wesley.

Tan, Z.-H., Dalsgaard, P., & Lindberg, B. (2007). Exploiting temporal correlation of speech for error-robust and bandwidth-flexible distributed speech recognition. *IEEE Transactions on Audio, Speech and Language Processing*, 15(4), 1391-1403.

Tranter, S., & Reynolds, D. (2006). An overview of automatic speaker diarization systems. *IEEE Transactions on Audio, Speech and Language Processing*, 14(5), 1557-1565.

Xu, H., Tan, Z.-H., Dalsgaard, P., & Lindberg, B. (2006). Robust speech recognition from noise-type based feature compensation and model interpolation in a multiple model framework. In *Proceedings of the 31st International Conference on Acoustics, Speech, and Signal Processing*.

Zhu, X., Wu, X., Elmagarmid, A.K., Feng, Z., & Wu, L. (2005). Video data mining: semantic indexing and event detection from the association perspective. *IEEE Transactions on Knowledge and Data Engineering*, 17(5), 665-677.

KEY TERMS

Audio Diarization: A process of segmenting an audio recording into homogeneous regions.

Audio Classification: A process of determining to which predefined class an audio file or segment belongs. It is fundamentally a pattern recognition problem.

Audio Clustering: A process of partitioning a set of audio files or segments into subsets or clusters such that audio content in each cluster share come common characteristics. This is done on the basis of some defined distance or similarity measure.

Automatic Speech Recognition: A process of converting a speech signal to a word sequence.

Metadata: A set of structured descriptions about data, or simply "data about data". Metadata is exploited to facilitate the management and use of data.

Pattern Discovery: A sub-discipline of data mining concerned with defining and detecting local anomalies in a given set of data, in contrast with modeling the entire data set.

Robust Speech Recognition: A field of research aimed at reinforcing the capability of speech recognition systems in coping well with variations in their operating environments.

Speech Data Mining: A process of extracting hidden and useful information from masses of speech data. In the process information like patterns, trends and anomalies are detected primarily for the purpose of decision making.

Audio Indexing

Gaël Richard
Ecole Nationale Supérieure des Télécommunications (TELECOM ParisTech), France

INTRODUCTION

The enormous amount of unstructured audio data available nowadays and the spread of its use as a data source in many applications are introducing new challenges to researchers in information and signal processing. The continuously growing size of digital audio information increases the difficulty of its access and management, thus hampering its practical usefulness. As a consequence, the need for content-based audio data parsing, indexing and retrieval techniques to make the digital information more readily available to the user is becoming ever more critical.

The lack of proper indexing and retrieval systems is making de facto useless significant portions of existing audio information (and obviously audiovisual information in general). In fact, if generating digital content is easy and cheap, managing and structuring it to produce effective services is clearly not. This applies to the whole range of content providers and broadcasters which can amount to terabytes of audio and audiovisual data. It also applies to the audio content gathered in private collection of digital movies or music files stored in the hard disks of conventional personal computers.

In summary, the goal of an audio indexing system will then be to automatically extract high-level information from the digital raw audio in order to provide new means to navigate and search in large audio databases. Since it is not possible to cover all applications of audio indexing, the basic concepts described in this chapter will be mainly illustrated on the specific problem of musical instrument recognition.

BACKGROUND

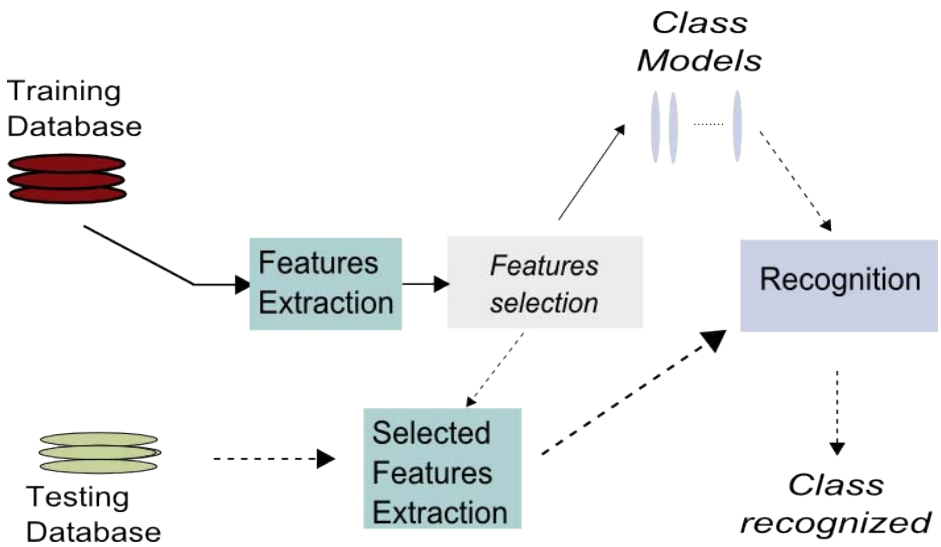
Audio indexing was historically restricted to word spotting in spoken documents. Such an application consists in looking for pre-defined words (such as name of a person, topics of the discussion etc…) in spoken documents by means of Automatic Speech Recognition (ASR) algorithms (see (Rabiner, 1993) for fundamentals of speech recognition). Although this application remains of great importance, the variety of applications of audio indexing now clearly goes beyond this initial scope. In fact, numerous promising applications exist ranging from automatic broadcast audio streams segmentation (Richard & et al., 2007) to automatic music transcription (Klapuri & Davy, 2006). Typical applications can be classified in three major categories depending on the potential users (Content providers, broadcasters or end-user consumers). Such applications include:

- Intelligent browsing of music samples databases for composition (Gillet & Richard, 2005), video scenes retrieval by audio (Gillet & et al., 2007) and automatic playlist production according to user preferences (for **content providers**).
- Automatic podcasting, automatic audio summarization (Peeters & et al., 2002), automatic audio title identification and smart digital DJing (for **broadcasters**).
- Music genre recognition (Tzanetakis & Cook, 2002), music search by similarity (Berenzweig & et al., 2004), personal music database intelligent browsing and query by humming (Dannenberg & et al. 2007) (for **consumers**).

MAIN FOCUS

Depending on the problem tackled different architectures are proposed in the community. For example, for musical tempo estimation and tracking traditional architectures will include a decomposition module which aims at splitting the signal into separate frequency bands (using a filterbank) and a periodicity detection module which aims at estimating the periodicity of a detection function built from the time domain envelope of the signal in each band (Scheirer, 1998)(Alonso & et al., 2007). When tempo or beat tracking is necessary, it will be coupled with onset detection techniques (Bello & et al., 2006) which aim at locating note onsets in

Figure 1. A typical architecture for a statistical audio indexing system based on a traditional bag-of frames approach. In a problem of automatic musical instrument recognition, each class represents an instrument or a family of instruments.



the musical signal. Note that the knowledge of note onset positions allows for other important applications such as Audio-to-Audio alignment or Audio-to-Score alignment.

However a number of different audio indexing tasks will share a similar architecture. In fact, a typical architecture of an audio indexing system includes two or three major components: A feature extraction module sometimes associated with a feature selection module and a classification or decision module. This typical "bag-of-frames" approach is depicted in Figure 1.

These modules are further detailed below.

Feature Extraction

The *feature extraction module* aims at representing the audio signal using a reduced set of features that well characterize the signal properties. The features proposed in the literature can be roughly classified in four categories:

- Temporal features: These features are directly computed on the time domain signal. The advantage of such features is that they are usually straightforward to compute. They include amongst others the crest factor, temporal centroid, zero-crossing rate and envelope amplitude modulation.
- Cepstral features: Such features are widely used in speech recognition or speaker recognition due to a clear consensus on their appropriateness for these applications. This is duly justified by the fact that such features allow to estimate the contribution of the filter (or vocal tract) in a source-filter model of speech production. They are also often used in audio indexing applications since many audio sources also obey a source filter model. The usual features include the Mel-Frequency Cepstral Coefficients (MFCC), and the Linear-Predictive Cepstral Coefficients (LPCC).
- Spectral features: These features are usually computed on the spectrum (magnitude of the Fourier Transform) of the time domain signal. They include the first four spectral statistical moments, namely the spectral centroid, the spectral width, the spectral asymmetry defined from the spectral skewness, and the spectral kurtosis describing the peakedness/flatness of the spectrum. A number of spectral features were also defined in the framework of MPEG-7 such as for example the MPEG-7 Audio Spectrum Flatness and Spectral Crest Factors which are processed over a number of frequency bands (ISO, 2001). Other features

proposed include the Spectral slope, the the spectral variation and the frequency cutoff. Some specific parameters were also introduced by (Essid & al. 2006a) for music instrument recognition to capture in a rough manner the power distribution of the different harmonics of a musical sound without recurring to pitch-detection techniques: the Octave Band Signal Intensities and Octave Band Signal Intensities Ratios.

- Perceptual features: Typical features of this class include the relative specific loudness representing a sort of equalization curve of the sound, the sharpness - as a perceptual alternative to the spectral centroid based on specific loudness measures- and the spread, being the distance between the largest specific loudness and the total loudness.

For all these features, it is also rather common to consider their variation over time through their first and second derivatives.

It is also worth to mention that due to their different dynamic it is often necessary to normalize each feature. A commonly used transformation scheme consists in applying a linear transformation to each computed feature to obtain centered and unit variance features. This normalization scheme is known to be more robust to outliers than a mapping of the feature dynamic range to a predefined interval such as [-1 : 1]. More details on most of these common features can be found in (Peeters, 2004) and in (Essid, 2005).

Features Selection

As mentioned above, when a large number of features is chosen, it becomes necessary to use *feature selection techniques* to reduce the size of the feature set (Guyon & Eliseeff, 2003). Feature selection techniques will consist in selecting the features that are the most discriminative for separating the different classes. A popular scheme is based on the Fisher Information Criterion which is expressed as the ratio of the inter-class spread to the intra-class spread. As such, a high value of the criterion for a given feature corresponds to a high separability of the class. The appropriate features can therefore be chosen by selecting those with the highest ratios.

Classification

The *classification module* aims at classifying or labelling a given audio segment. This module usually needs a training step where the characteristics of each class are learned. Popular supervised classification approaches for this task include K-nearest neighbours, Gaussian Mixture Models, Support Vector Machines (SVM) and Hidden Markov models (Burges, 1998), (Duda & al., 2000).

For example, in a problem of automatic musical instrument recognition (Essid & al., 2006a), a state of the art system will compute a large number of features (over 500), use feature selection and combine multiple binary SVM classifiers. When a large number of instruments is considered (or when polyphonic music involving more than one instrument playing at a time, as in (Eggink and Brown, 2004)), hierarchical approaches aiming first at recognising an instrument family (or group of instruments) are becoming very efficient (Essid & al. 2006b).

FUTURE TRENDS

Future trends in audio indexing are targeting robust and automatic extraction of high level semantic information in polyphonic music signals. Such information for a given piece of music could include the main melody line; the musical emotions carried by the musical piece, its genre or tonality; the number and type of musical instruments that are active. All these tasks which have already interesting solutions for solo music (e.g. for mono-instrumental music) become particularly difficult to solve in the context of real recordings of polyphonic and multi-instrumental music. Amongst the interesting directions, a promising path is provided by methods that try to go beyond the traditional "bag-of-frames" approach described above. In particular, sparse representation approaches that rely on a signal model (Leveau & al. 2008) or techniques based on mathematical decomposition such as Non-Negative Matrix factorization (Bertin & al. 2007) have already obtained very promising results in Audio-to-Score transcription tasks.

CONCLUSION

Nowadays, there is a continuously growing interest of the community for audio indexing and Music Information Retrieval (MIR). If a large number of applications already exist, this field is still in its infancy and a lot of effort is still needed to bridge the "semantic gap" between a low-level representation that a machine can obtain and the high level interpretation that a human can achieve.

REFERENCES

M. Alonso, G. Richard and B. David (2007) "Accurate tempo estimation based on harmonic+noise decomposition", *EURASIP Journal on Advances in Signal Processing*, vol. 2007, Article ID 82795, 14 pages. 2007.

J. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. Sandler, (2005) "A tutorial on onset detection in musical signals," *IEEE Trans. Speech and Audio Processing*, vol. 13, no. 5, pp. 1035–1047. 2005

A. Berenzweig, B. Logan, D. Ellis, B. Whitman (2004). A large-scale evaluation of acoustic and subjective music-similarity measures, *Computer Music Journal*, 28(2), pp. 63-76, June 2004.

N. Bertin, R. Badeau and G. Richard, (2007) "Blind signal decompositions for automatic transcription of polyphonic music: NMF and K-SVD on the benchmark", *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'07*, Honolulu, Hawaii, USA, 15-20 april 2007.

C. J. Burges, (1998) "A tutorial on support vector machines for pattern recognition," *Journal of Data Mining and knowledge Discovery*, vol. 2, no. 2, pp. 1–43, 1998.

R. Dannenberg, W. Birmingham, B. Pardo, N. Hu, C. Meek and G. Tzanetakis, (2007) "A comparative evaluation of search techniques for query by humming using the MUSART testbed." *Journal of the American Society for Information Science and Technology* 58, 3, Feb. 2007.

R. Duda, P. Hart and D. Stork, (2000) *Pattern Classification,*. Wiley-Interscience. John Wiley and Sons, (2nd Edition) 2000.

J. Eggink and G. J. Brown, (2004) "Instrument recognition in accompanied sonatas and concertos",. in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP*), Montreal, Canada, May 2004, pp. 217.220.

S. Essid, G. Richard and B. David, (2006) "Musical Instrument Recognition by pairwise classification strategies", *IEEE Transactions on Speech, Audio and Language Processing*, Volume 14, Issue 4, July 2006 Page(s):1401 - 1412.

S. Essid, G. Richard and B. David, (2006), "Instrument recognition in polyphonic music based on automatic taxonomies", *IEEE Transactions on Audio, Speech, and Language Processing,* Vol. 14, N. 1, pp. 68-80

S. Essid, (2005) *Automatic Classification of Audio Signals: Machine Recognition of Musical Instruments*. PhD thesis ,Université Pierre et Marie Curie. December 2005 (In French)

O. Gillet, S. Essid and G. Richard, (2007) "On the Correlation of Automatic Audio and Visual Segmentations of Music Videos", *IEEE Transaction On Circuit and Systems for Video Technology,* Vol. 17, N. 3, March 2007.

O. Gillet and G. Richard, (2005) "Drum loops retrieval from spoken queries", *Journal of Intelligent Information Systems - Special issue on Intelligent Multimedia Applications*, vol. 24, n° 2/3, pp. 159-177, March 2005.

I. Guyon and A. Elisseeff, (2003) An introduction to feature and variable selection,. *Journal of Machine Learning Research*, vol. 3, pp. 1157.1182, 2003.

ISO, (2001). Information technology - multimedia content description interface - part 4: Audio,. ISO/IEC, International Standard ISO/IEC FDIS 15938-4:2001(E), jun 2001.

A. Klapuri and M. Davy, editors. (2006) *Signal Processing methods for the automatic transcription of music*. Springer, New-York, 2006.

D.D. Lee and H.S. Seung, (2001) Algorithms for non-negative matrix factorization, *Advances in Neural Information Processing Systems*, vol. 13, pp. 556–562, 2001.

P. Leveau, E. Vincent, G. Richard, L. Daudet. (2008) Instrument-specific harmonic atoms for mid-level music representation. *To appear in IEEE Trans. on Audio, Speech and Language Processing*, 2008.

G. Peeters, A. La Burthe, X. Rodet, (2002) Toward Automatic Music Audio Summary Generation from Signal Analysis, in *Proceedings of the International Conference of Music Information Retrieval (ISMIR)*, 2002.

G. Peeters, (2004) "A large set of audio features for sound description (similarity and classification) in the cuidado project," *IRCAM, Technical Report*, 2004.

L. R. Rabiner, (1993) *Fundamentals of Speech Processing*, ser. Prentice Hall Signal Processing Series. PTR Prentice-Hall, Inc., 1993.

G. Richard, M. Ramona and S. Essid, (2007) "Combined supervised and unsupervised approaches for automatic segmentation of radiophonic audio streams, *in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Honolulu, Hawaii, 2007.

E. D. Scheirer. (1998) Tempo and Beat Analysis of Acoustic Music Signals. *Journal of Acoustical Society of America*, 103 :588-601, janvier 1998.

G. Tzanetakis and P. Cook, (2002) Musical genre classification of audio signals,. *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, July 2002.

KEY TERMS

Features: Features aimed at capturing one or several characteristics of the incoming signal. Typical features include the energy, the Mel-frequency cepstral coefficients.

Frequency Cutoff (or Roll-off): Computed as the frequency below which 99% of the total spectrum energy is concentrated.

Mel-Frequency Cepstral Coefficients (MFCC): are very common features in audio indexing and speech recognition applications. It is very common to keep only the first few coefficients (typically 13) so that they mostly represent the spectral envelope of the signal.

Musical Instrument Recognition: The task to automatically identify from a music signal which instruments are playing. We often distinguish the situation where a single instrument is playing with the more complex but more realistic problem of recognizing all instruments of real recordings of polyphonic music.

Non-Negative Matrix Factorization: This technique permits to represent the data (e.g. the magnitude spectrogram) as a linear combination of elementary spectra, or atoms and to find from the data both the decomposition and the atoms of this decomposition (see [Lee & al., 2001] for more details).

Octave Band Signal Intensities: These features are computed as the log-energy of the signal in overlapping octave bands.

Octave Band Signal Intensities Ratios: These features are computed as the logarithm of the energy ratio of each subband to the previous (e.g. lower) subband.

Semantic Gap: Refers to the gap between the low-level information that can be easily extracted from a raw signal and the high level semantic information carried by the signal that a human can easily interpret.

Sparse Representation Based on a Signal Model: Such methods aim at representing the signal as an explicit linear combination of sound sources, which can be adapted to better fit the analyzed signal. This decomposition of the signal can be done using elementary sound templates of musical instruments.

Spectral Centroid: Spectral centroid is the first statistical moment of the magnitude spectrum components (obtained from the magnitude of the Fourier transform of a signal segment).

Spectral Slope: Obtained as the slope of a line segment fit to the magnitude spectrum.

Spectral Variation: Represents the variation of the magnitude spectrum over time.

Support Vector Machines: Support Vector Machines (SVM) are powerful classifiers arising from Structural Risk Minimization Theory that have proven to be efficient for various classification tasks, including speaker identification, text categorization and musical instrument recognition.

An Automatic Data Warehouse Conceptual Design Approach

Jamel Feki
Mir@cl Laboratory, Université de Sfax, Tunisia

Ahlem Nabli
Mir@cl Laboratory, Université de Sfax, Tunisia

Hanêne Ben-Abdallah
Mir@cl Laboratory, Université de Sfax, Tunisia

Faïez Gargouri
Mir@cl Laboratory, Université de Sfax, Tunisia

INTRODUCTION

Within today's competitive economic context, information acquisition, analysis and exploitation became strategic and unavoidable requirements for every enterprise. Moreover, in order to guarantee their persistence and growth, enterprises are forced, henceforth, to capitalize expertise in this domain.

Data warehouses (DW) emerged as a potential solution answering the needs of storage and analysis of large data volumes. In fact, a DW is a database system specialized in the storage of data used for decisional ends. This type of systems was proposed to overcome the incapacities of OLTP (On-Line Transaction Processing) systems in offering analysis functionalities. It offers integrated, consolidated and temporal data to perform decisional analyses. However, the different objectives and functionalities between OLTP and DW systems created a need for a development method appropriate for DW.

Indeed, data warehouses still deploy considerable efforts and interests of a large community of both software editors of decision support systems (DSS) and researchers (Kimball, 1996; Inmon, 2002). Current software tools for DW focus on meeting end-user needs. OLAP (On-Line Analytical Processing) tools are dedicated to multidimensional analyses and graphical visualization of results (*e.g.*, Oracle Discoverer®); some products permit the description of DW and Data Mart (DM) schemes (*e.g.*, Oracle Warehouse Builder®). One major limit of these tools is that the schemes must be built beforehand and, in most cases, manually. However, such a task can be tedious, error-prone and time-consuming, especially with heterogeneous data sources.

On the other hand, the majority of research efforts focuses on particular aspects in DW development, *cf.*, multidimensional modeling, physical design (materialized views (Moody & Kortnik, 2000), index selection (Golfarelli, Rizzi, & Saltarelli 2002), schema partitioning (Bellatreche & Boukhalfa, 2005)) and more recently applying data mining for a better data interpretation (Mikolaj, 2006; Zubcoff, Pardillo & Trujillo, 2007). While these practical issues determine the performance of a DW, other just as important, conceptual issues (*e.g.*, requirements specification and DW schema design) still require further investigations. In fact, few propositions were put forward to assist in and/or to automate the design process of DW, *cf.*, (Bonifati, Cattaneo, Ceri, Fuggetta & Paraboschi, 2001; Hahn, Sapia & Blaschka, 2000; Phipps & Davis 2002; Peralta, Marotta & Ruggia, 2003).

This chapter has a twofold objective. First, it proposes an intuitive, tabular format to assist decision maker in formulating their OLAP requirements. It proposes an automatic approach for the conceptual design of DW/DM schemes, starting from specified OLAP requirements. Our automatic approach is composed of four steps: *Acquisition of OLAP requirements*, *Generation of star/constellation schemes*, *DW schema generation*, and *Mapping the DM/DW onto data sources*. In addition, it relies on an algorithm that transforms tabular OLAP requirements into DM modelled either as a star or a constellation schema. Furthermore, it applies a set of mapping rules between the data sources and the DM schemes. Finally, it uses a set of unification rules

that merge the generated DM schemes and construct the DW schema.

BACKGROUND

There are several proposals to automate certain tasks of the DW design process (Hahn, Sapia & Blaschka, 2000). In (Peralta, Marotta & Ruggia, 2003), the authors propose a rule-based mechanism to automate the construction of the DW logical schema. This mechanism accepts the DW conceptual schema and the source databases. That is, it supposes that the DW conceptual schema already exists. In addition, being a bottom-up approach, this mechanism lacks a conceptual design methodology that takes into account the user requirements which are crucial in the DW design.

In (Golfarelli, Maio & Rizzi, 1998), the authors propose how to derive a DW conceptual schema from Entity-Relationship (E/R) schemes. The conceptual schema is represented by a Dimensional-Fact Model (DFM). In addition, the translation process is left to the designer, with only interesting strategies and cost models presented. Other proposals, similar to (Marotta & Ruggia 2002; Hahn, Sapia & Blaschka, 2000) also generate star schemes and suppose that the data sources are E/R schemes. Although the design steps are based on the operational data sources, the end-users' requirements are neglected. Furthermore, the majority of these works use a graphical model for the Data Source (DS) from which they generate the DM schema; that is, they neither describe clearly how to obtain the conceptual graphical models from the DS, nor how to generate the multidimensional schemes.

Other works relevant to automated DW design mainly focus on the conceptual design, *e.g.*, (Hüsemann, Lechtenbörger & Vossen 2000) and (Phipps & Davis 2002) who generate the conceptual schema from an E/R model. However, these works do not focus on a conceptual design methodology based on users' requirements and are, in addition, limited to the E/R DS model.

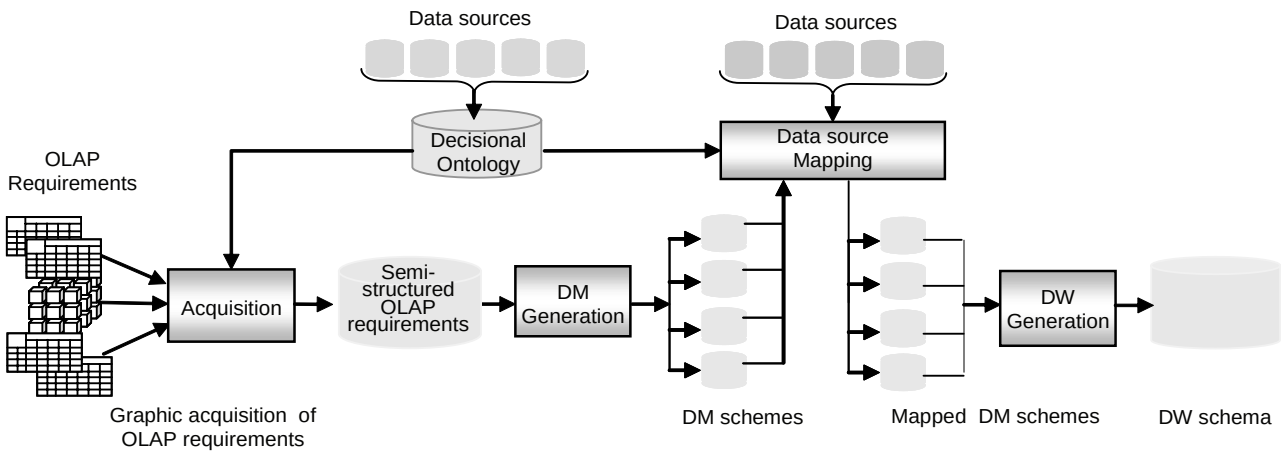
MAIN FOCUS

We propose an automatic approach to design DW/DM schemes from precisely specified OLAP requirements. This approach (Figure 1) is composed of four steps: *i) Acquisition of OLAP requirements* specified as two/n-dimensional fact sheets producing "semi-structured" OLAP requirements, *ii) Generation of star/constellation schemes* by merging the semi-structured OLAP requirements, *iii) DW generation schema* by fusion of DM schemes, and *iv) Mapping the DM/DW* to the data sources.

OLAP Requirement Acquisition

Decisional requirements can be formulated in various manners, but most generally they are expressed in natural language sentences. In our approach, which aims at a computer aided design, we propose to collect these requirements in a format familiar to the decision

Figure 1. DW design starting from OLAP requirements



makers: structured sheets. As illustrated in Figure 2, our generic structure defines the *fact* to be analyzed its *domain*, its *measures* and its analysis *dimensions*. We call this structure "2D-F sheet", acronym for Two-Dimensional Fact sheet. To analyze a fact with *n* (*n*>2) dimensions, we may need to use several 2D-F sheets simultaneously or hide one dimension at a time to add a new one to the sheet to obtain *nD-F* sheet. With this format, the OLAP requirements can be viewed as a set of 2/nD-F sheets, each defining a fact and two/n analysis dimensions.

We privileged this input format because it is familiar and intuitive to decision makers. As illustrated in Figure 1, the requirement acquisition step uses a decisional ontology specific to the application domain. This ontology supplies the basic elements and their multidimensional semantic relations during the acquisition. It assists the decisional user in formulating their needs and avoiding naming and relational ambiguities of dimensional concepts (Nabli, Feki & Gargouri 2006).

Example: Figure 3 depicts a 2D-F sheet that analyzes the *SALE* fact of the *commercial* domain. The measure *Qty* depends of the dimensions *Client* and *Date*.

The output of the acquisition step is a set of sheets defining the facts to be analyzed, their measures and dimensions, dimensional attributes, etc. These specified requirements, called semi-structured OLAP requirements, are the input of the next step: DM generation.

DM Schema Generation

A DM is subject-oriented and characterized by its multidimensional schema made up of facts measured along analysis dimensions. Our approach aims at constructing the DM schema starting from OLAP requirements specified as a set of 2/nD-F sheets. Each sheet can be seen as a partial description (i.e., multidimensional view) of a DM schema. Consequently, for a given domain, the complete multidimensional schemes of the DMs are derived from all the sheets specified in the acquisition step. For this, we have defined a set of algebraic operators to derive automatically the DM schema (Nabli, Feki & Gargouri, 2005).

This derivation is done in two complementary phases according to whether we want to obtain star or constellation schemes:

1. *Generation of star schemes*, which groups sheets referring to the same domain and describing the same fact. It then merges all the sheets identified within a group to build a star.

Figure 2. Generic structure of 2D-F sheet

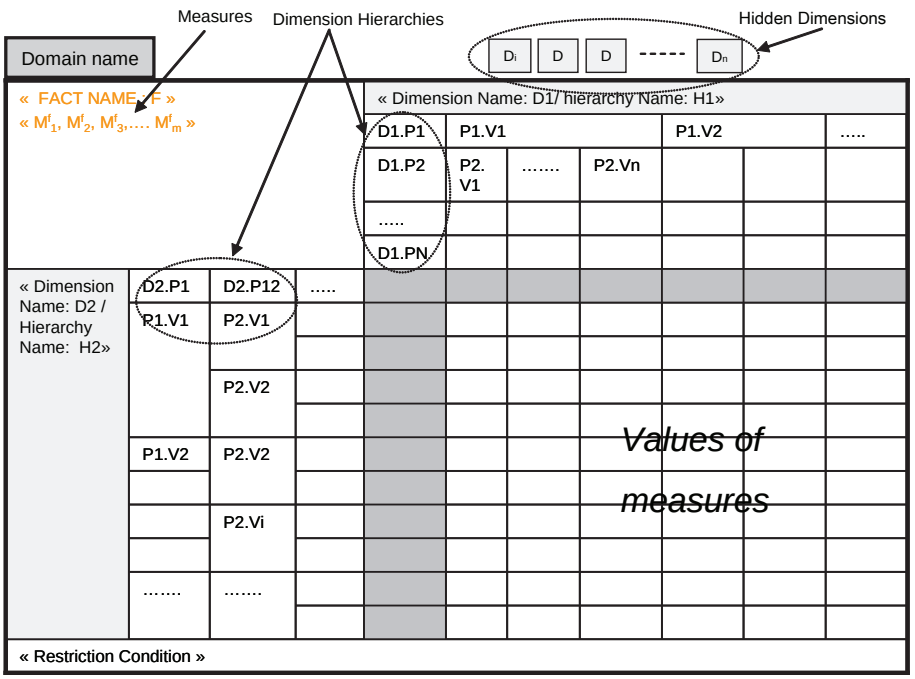
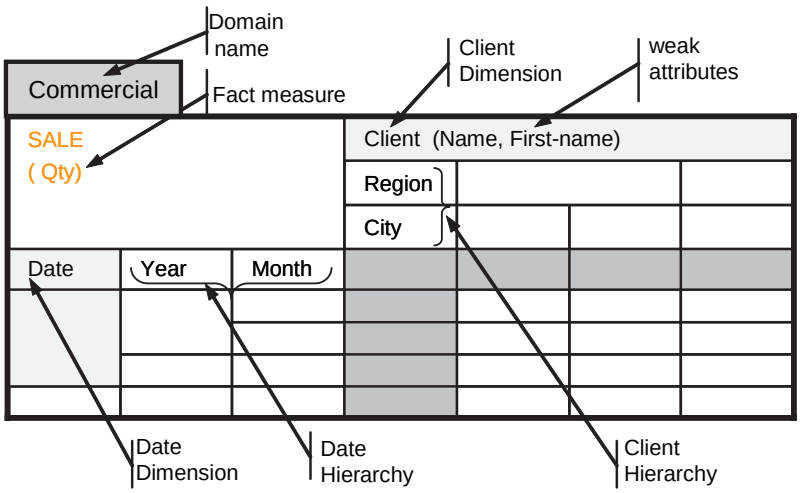


Figure 3. T1. 2D-F sheet for the SALE fact



Box 1.

Star_Generation
Begin
Given *t* *n*D-F sheets analyzing *f* facts belonging to *m* analysis domains (*m*<=*t*).
Partition the *t* sheets into the *m* domains, to obtain G_{dom1}, G_{dom2},….., G_{domm} sets of sheets.
For each G_{domi} (i=1..*m*)
Begin
1.1. Partition the sheets in G_{domi} by facts into G^{F1}_{domi},…… , G^{Fk}_{domi} (k<=f)
1.2. For each G^{Fj}_{domi} (j=1..k)
Begin
1.2.1. For each sheet s ∈ G^{Fj}_{domi}
For each dimension d ∈ dim(s)
Begin
- Complete the hierarchy of d to obtain a maximal hierarchy.
- Add an identifier Id^d as an attribute.
End
1.2.2. Collect measures $Mes^{Fj} = \bigcup_{s \in G^{Fj}_{domi}} meas(s)$
1.2.3. Create the structure of a fact F for F_j with Mes^{Fj}.
1.2.4. Collect dimensions $Dim^{Fj} = \bigcup_{s \in G^{fj}_{domi}} \dim(s)$
1.2.4.1. For each d ∈ Dim^{Fj}
Begin
- Determine hierarchies $hier_d^{Fj} = \bigcup_{s \in G^{Fj}_{domi}} \bigcup_{d \in \dim(s)} hier(d)$
- Create the structure of a dimension D for d with $hier^{Fj}_d$.
- Associate D with F.
End
End
End.

2. *Generation of constellation schemes,* which integrates star schemes relevant to the same domain and that may have common dimensions.

To present the algorithms of these two types of schema generation, we will use the following notation.

– *Dim(s):* the set of dimensions in an nD-F sheet *s*,
– *Hier(d):* the hierarchy of a dimension *d*,
– *Meas(s):* the set of measures in an nD-F sheet *s*.

Star Schema Generation

The algorithm shown in Box 1 (Nabli, Soussi, Feki, Ben-Abdallah & Gargouri, 2005) generates star schemes. In this algorithm, the *t* nD-F sheets are first partitioned into domains; this ensures that each star schema is generated for one domain. In turn, this will reduce the number of comparisons used in the constellation schema generation phase (see next section). A star schema is constructed for each fact (F_j) in steps 3.2.2. to 3.2.5.

Example: Let us extend the previous *SALE* example with the two additional sheets shown in Figure 4. The

Figure 4. T2 and T3 two sheets for the SALE fact

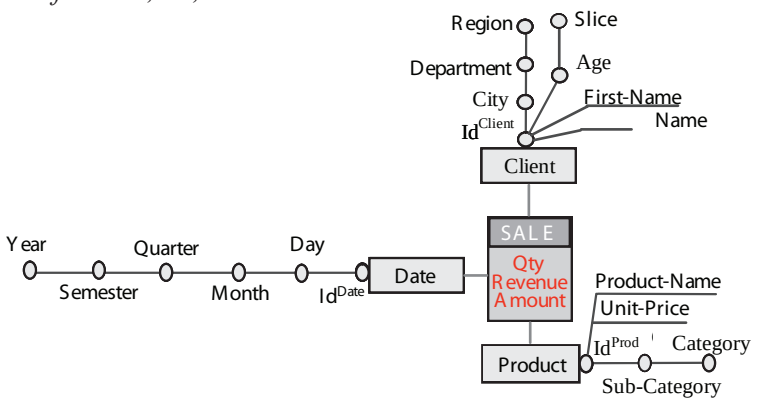
Commercial **T2**

SALE (Qty, Amount)				Product (unit-price, prod-name) /H_category			
				category			
Date H_Year	Year	Quarter	Month				

Commercial **T3**

SALE (Qty, Revenue)				Client (Name, First-name)/H_age			
				Age			
				Slice			
Date H_Year	Year	Semester	Month				

Figure 5. Star schema built from T1, T2, and T3 sheets



star schema resulting from applying our algorithm is shown in Figure 5.

Note that Id^{Date}, Id^{Client} and Id^{Prod} were added as attributes to identify the dimensions. In addition, several attributes were added to complete the dimension hierarchies. This addition was done in step 3.2.1. of the algorithm.

Constellation Schema Generation

In the above phase, we have generated a star schema for each fact in the same analysis domain. These latter have to be merged to obtain star/constellation schemes. For this, we adapt the *similarity factor* of (Feki, 2004) to measure the pertinence of schemes to be integrated, *i.e.*, the number of their common dimensions.

Given S_i and S_k two star schemes in the same analysis domain, their similarity factor $Sim(S_i, S_k)$ is calculated on the basis of n and m which are the number of dimensions in S_i and S_k respectively, and p which is the number of their common dimensions:

$$Sim(S_i, S_K) = \begin{cases} \alpha \quad if\,(n = p) \wedge (n < m); \\ p\,/\,(n + m - p)\; otherwise. \end{cases}$$

Informally, $Sim(S_i, S_k)$ highlights the case where all the dimensions of S_i are included in S_k. To dismiss the trivial case of S_i is having only the *Date* dimension (present in all schemes), the designer should fix the threshold α to a value strictly greater than 0.5.

In addition, to enhance the quality of the integration result, we define a matrix of similarities *MS* to measure the similarity between each pair of multidimensional schemes. This matrix is used to decide which schemes should be integrated first.

Given n star schemes of the same analysis domain $S_1, S_2, \ldots, S_n$. Each schema, defined by a name, analyzes a fact and has a set of dimensions. The *stopcondition* is a Boolean expression, *true* if either the size of *MS* becomes 1 or all the values in *MS* are lower than a threshold set by the designer. Let us extend our previous example with the additional star S2 (Figure 6).

The five steps of the DM schema construction are:

a. Construct the matrix of similarities *MS*
b. Find all the occurrences of the maximum *max* in *MS*
c. Construct a constellation by merging all schemes having the maximum similarity *max*
d. Re-dimension *MS* by:
   - Dropping rows and columns of the merged schemes
   - Adding one row and one column for the newly constructed schema
e. If <*stopcondition*> then exit, else return to step a.

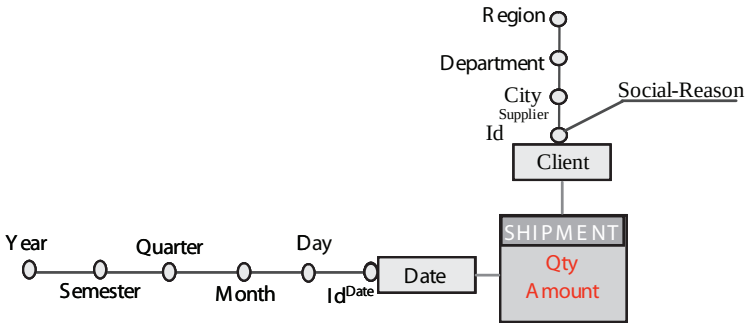
The similarity matrix for *S1* and *S2* contains the single value *Sim(S1, S2) = 0.75.*

The constellation schema resulting from applying the above five steps is depicted by Figure 7.

DM-DS Mapping

The DW is built from several data sources (DS) while its schema is built from the DM schemes. Thus, the DM schemes must be mapped to the DS schemes. In our approach, the DM-DS mapping adapts the heuristics proposed by (Golfarelli, Maio & Rizzi , 1998; Bonifati, Cattaneo, Ceri, Fuggetta & Paraboschi, 2001) to map each element (i.e., fact, dimension…) of the DM

Figure 6. S2 star schema



schemes to one or more elements (entity, relation, attribute…) of the DS schemes.

Our DM-DS mapping is performed in three steps: first, it identifies from the DS schema potential facts (PF), and matches facts in the DM schemes with identified PF. Secondly, for each mapped fact, it looks for DS attributes that can be mapped to measures in the DM schemes. Finally, for each fact that has potential measures, it searches DS attributes that can be mapped to dimensional attributes in the DM schemes.

A DM element may be derived from several identified potential elements. In addition, the same element can be mapped to several identified potential elements. It may also happen that a DM element is different from all potential elements, which might require OLAP requirement revision.

Fact Mapping

Fact mapping aims to find for each DM fact (Fd) the corresponding DS elements. For this, we first identify DS elements that could represent potentiel facts (PF). Then, we confront the set of Fd with all identified PF. The result of this step is a set of (Fd, PF) pairs for which the measures and dimensions must be confronted to accept or reject the mapping (cf. validation mapping step).

Fact Identification

Each entity of the DS verifying one of the following two rules becomes a potential fact:

F1: An *n-ary* relationship in the DS with numerical attribute with $n \geq 2$;

F2: An entity with at least one numerical attribute not included in its identifier.

Fact Matching

An ontology dedicated to decisional system is used to find for each fact in the DM schema all corresponding potential facts. In this step, we may encounter one problematic case: a DM fact has no corresponding PF. Here the designer must intervene.

Note that, when a DM fact has several corresponding PF, all mappings are retained until the measures and dimensions are identified.

Measure Mapping

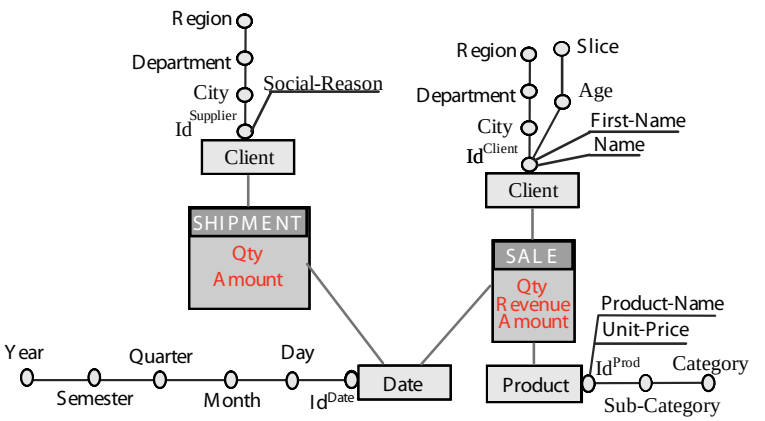
For each (Fd, PF) determined in the previous step, we identify the potential measures of PF and confront them with those of Fd.

Measure Identification

Since measures are numerical attributes, they will be searched within potential facts (PF) and "parallel" entities; they will be qualified as potential measures (PM). The search order is:

1. A non-key numerical attribute of PF
2. A non-key numerical attribute of parallel entities to PF
3. A numerical attribute of the entities related to PF by a "one-to-one" link first, followed by those related to PF by a "one-to-many" link
4. Repeat step 3 for each entity found in step 3

Figure 7. Constellation schema built from the stars in Figures 5 and 6



Measure Matching
Given the set of potential measures of each PF, we use a decisional ontology to find the corresponding measures in Fd. A DM measure may be derived from several identified PM. The identified PM that are matched to fact Fd measures are considered the measures of the PF.

In this step, we eliminate all (Fd, PF) for which no correspondence between their measures is found.

Dimension Mapping
This step identifies potential dimension attributes and confronts them with those of Fd, for each (Fd, PF) retained in the measure mapping phase.

Dimension Attribute Identification
Each attribute, not belonging to any potential measures and verifying the following two rules, becomes a potential dimension (PD) attribute:

D1: An attribute in a potential fact PF
D2: An attribute of an entity related to a PF via a "one-to-one" or "one-to-many" link. The entity relationships take into account the transitivity

Note that, the order in which the entities are considered determines the hierarchy of the dimension attributes. Thus, we consult the attributes in the following order:

1. An attribute of PF, if any
2. An attribute of the entities related to PF by a "one-to-one" link initially, followed by the attributes of the entities related to PF by "one-to-many" link
3. Repeat step 2 for each entity found in step 2

Dimension Matching
Given the set of PD attribute, we use a decisional ontology to find the corresponding attribute in Fd. If we can match the identifier of a dimension *d* with a PD attributes, this later is considered as a PD associated to PF.

In this step, we eliminate all (Fd, PF) for which no correspondence between their dimensions is found.

DM-DS Mapping Validation
The crucial step is to specify how the ideal requirements can be mapped to the real system. The validation may also give the opportunity to consider new analysis aspects that did not emerge from user requirements, but that the system may easily make available. When a DM has one corresponding potential fact, the mapping is retained. Whereas, when a DM fact has several corresponding potential facts {(Fd, PF)}, the measures of Fd are the union of the measures of all PF. This is argued by the fact that the identification step associates each PM with only one potential fact; hence, all sets of measures are disjoint. Multiple correspondences of dimensions are treated in the same way.

Data Warehouse Generation

Our approach distinguishes two storage spaces: the DM and the DW designed by two different models. The DMs have multidimensional models, to support OLAP analysis, whereas the DW is structured as a conventional database. We found the UML class diagram appropriate to represent the DW schema.

The DM schema integration is accomplished in the DW generation step (see Figure 1) that operates in two complementary phases:

1. Transform each DM schema (i.e. stars and constellations) into an UML class diagram.
2. Merge the UML class diagrams. This merger produces the DW schema independent of any data structure and content.

Recall that a dimension is made up of hierarchies of attributes. The attributes are organized from the finest to the highest granularity. Some attributes belong to the dimension but not to hierarchies; these attributes are called weak attributes, they serve to label results.

The transformation of DM schemes to UML class diagrams uses a set of rules among which we list the following five rules (For further details, the reader is referred to Feki, 2005):

Rule 1: *Transforming a dimension d into classes* − Build a class for every non-terminal attribute of each hierarchy of d.
Rule 2: *Assigning attributes to classes* − A class built from an attribute *a* gathers this attribute, the weak attributes associated to *a*, and the terminal attributes that are immediately related to *a* and not having weak attributes.
Rule 3: *Linking classes* − Each class *Ci* built from attribute at level *i* of a hierarchy *h*, is con-

nected via a composition link to the class C_{i-1}, of the same hierarchy, if any.

Rule 4: *Transforming facts into associations* − A fact table is transformed into an association linking the finest level classes derived from its dimensions. Measures of the fact become attributes of the association.

Note that the above four rules apply only to non-date dimension. Rule 5 deals with the date dimension:

Rule 5: *Transforming date dimension* – A date dimension is integrated into each of its related fact classes as a full-date, *i.e.*, detailed date.

FUTURE TRENDS

We are currently verifying the completeness of the set of DM to DW schema transformation rules; the proof proceeds by induction on the schema structure. In addition, we are examining how to adapt our approach to a model-driven development approach like MDA (Model Driven Architecture) of OMG (OMG, 2002) (Mazón & Trujillo, 2007). Such an alignment will allow us to formalize better the transformations among the models. In addition, it can benefit from the decisional ontology as a starting Computation Independent Model (CIM). The decisional end-user instantiates the decisional elements from this ontology in order to formulate their particular requirements as nD-F; thus, the nD-F can be regarded as a form of Platform Independent Model (PIM). This later can be transformed, through our set of transformations, to derive a DM/DW schema.

CONCLUSION

This work lays the grounds for an automatic, systematic approach for the generation of data mart and data warehouse conceptual schemes. It proposed a standard format for OLAP requirement acquisition, and defined an algorithm that transforms automatically the OLAP requirements into multidimensional data mart schemes. In addition, it outlined the mapping rules between the data sources and the data marts schemes. Finally, it defined a set of unification rules that merge the generated data mart schemes to construct the data warehouse schema.

REFERENCES

Bellatreche, L., & Boukhalfa, K. (2005). An evolutionary approach to schema partitioning selection in a data warehouse. *7th International Conference on Data Warehousing and Knowledge Discovery*. Springer-Verlag.

Bonifati, A., Cattaneo, F., Ceri, S., Fuggetta, A., & Paraboschi, S. (2001). Designing data marts for data warehouses. *ACM Transactions on Software Engineering Methodology*.

Feki, J. (2004). Vers une conception automatisée des entrepôts de données : Modélisation des besoins OLAP et génération de schémas multidimensionnels. *8th Maghrebian Conference on Software Engineering and Artificial Intelligence*, 473-485, Tunisia.

Feki, J., Majdoubi, J., & Gargouri, F. (2005). A two-phase approach for multidimensional schemes integration. *The 7th International Conference on Software Engineering and Knowledge Engineering*.

Golfarelli, M., Maio, D., & Rizzi, S. (1998). Conceptual design of data warehouses from E/R schemes. *Hawaii International Conference on System Sciences*.

Golfarelli, M., Rizzi, S., & Saltarelli, E. (2002). Index selection for data warehousing. *Design and Management of Data Warehouses*, 33-42.

Hahn, K., Sapia, C., & Blaschka, M. (2000). automatically generating OLAP schemes from conceptual graphical models. *International Workshop on Data Warehousing and OLAP*.

Hüsemann, B., Lechtenbörger J. & Vossen G. (2000). Conceptual data warehouse design. *Design and Management of Data Warehouses*, Sweden.

Inmon, W. H. (2002). *Building the data warehouse*. Wiley Press.

Kimball, R. (1996). *The data warehouse toolkit*. New York: John Wiley and Sons, Inc.

Marotta, A., & Ruggia, R. (2002). Data warehouse design: A schema-transformation approach. *International Conference of the Chilean Computer Science Society*, 153-161, Chile.

Mazón, J.-N., & Trujillo, J. (2007). An MDA approach for the development of data warehouses. *Decision Support Systems*.

Mikolaj, M. (2006). Efficient mining of dissociation rules. *Data Warehousing and Knowledge Discovery*.

Moody, D., & Kortnik, M. (2000). From enterprise models to dimensionals models: A methodology for data warehouse and data mart design. *Design and Management of Data Warehouses*.

Nabli, A., Feki, J., & Gargouri, F. (2005). Automatic construction of multidimensional schema from OLAP requirements. *ACS/IEEE International Conference on Computer Systems and Applications*.

Nabli, A., Feki, J., & Gargouri, F. (2006). An ontology based method for normalisation of multidimensional terminology. *Signal-Image Technology and Internet-based Systems*.

Nabli, A., Soussi, A.,Feki, J., Ben-Abdallah, H. & Gargouri, F. (2005). Towards an automatic data mart design. *Seventh International Conference on Enterprise Information Systems,* 226-231.

Object Management Group (2002). *The common warehouse metamodel (CWM)*. http://www.omg.org/cgibin/doc?formal/03-03-02

Peralta, V., Marotta, A., & Ruggia, R. (2003). *Towards the automation of data warehouse design*. Technical Report, Universidad de la República, Uruguay.

Phipps, C., & Davis, K. (2002). Automating data warehouse conceptual schema design and evaluation. *Design and Management of Data Warehouses*.

Zubcoff, J. J., Pardillo, J. & Trujillo, J. (2007). Integrating clustering data mining into the multidimensional modeling of data warehouses with UML profiles. *Data Warehousing and Knowledge Discovery*, 199-208.

KEY TERMS

Decisional Ontology: A decisional ontology is a representation of knowledge dedicated to the decisional systems. It is a referential of multidimensional concepts of a field, their semantic and multidimensional relations.

Maximal Hierarchy: A hierarchy is called *maximal* if it cannot be extended upwards or downwards by including another attribute.

Multidimensional Model: data are modelled as dimensional schemes composed of a set of facts, dimensions and hierarchies. It can be either a star or a constellation.

OLAP Requirement Model: a tabular, two/n-dimensional fact sheet (2/nD-F) that describes a fact F in a domain, its measures, and its two (n) dimensions of analysis.

Parallel Entities: Two entities *E1* and *E2* are "parallel" if the set of entities related to *E1* by a one-to-one link is included in the set of entities related to *E2* by one-to-one links.

Schema Integration: Merges multidimensional schemes with a high similarity factor in order to build a constellation schema that enables drill across analyses.

Similarity Factor: a ratio that reflects the number of common dimensions between two multidimensional schemes.

Automatic Genre-Specific Text Classification

Xiaoyan Yu
Virginia Tech, USA

Manas Tungare
Virginia Tech, USA

Weiguo Fan
Virginia Tech, USA

Manuel Pérez-Quiñones
Virginia Tech, USA

Edward A. Fox
Virginia Tech, USA

William Cameron
Villanova University, USA

Lillian Cassel
Villanova University, USA

INTRODUCTION

Starting with a vast number of unstructured or semi-structured documents, text mining tools analyze and sift through them to present to users more valuable information specific to their information needs. The technologies in text mining include information extraction, topic tracking, summarization, categorization/classification, clustering, concept linkage, information visualization, and question answering [Fan, Wallace, Rich, & Zhang, 2006]. In this chapter, we share our hands-on experience with one specific text mining task — text classification [Sebastiani, 2002].

Information occurs in various formats, and some formats have a specific structure or specific information that they contain: we refer to these as *`genres'*. Examples of information genres include news items, reports, academic articles, etc. In this paper, we deal with a specific genre type, course syllabus.

A course syllabus is such a genre, with the following commonly-occurring fields: title, description, instructor's name, textbook details, class schedule, etc. In essence, a course syllabus is the skeleton of a course. Free and fast access to a collection of syllabi in a structured format could have a significant impact on education, especially for educators and life-long learners. Educators can borrow ideas from others' syllabi to organize their own classes. It also will be easy for life-long learners to find popular textbooks and even important chapters when they would like to learn a course on their own. Unfortunately, searching for a syllabus on the Web using Information Retrieval [Baeza-Yates & Ribeiro-Neto, 1999] techniques employed by a generic search engine often yields too many non-relevant search result pages (i.e., noise) — some of these only provide guidelines on syllabus creation; some only provide a schedule for a course event; some have outgoing links to syllabi (e.g. a course list page of an academic department). Therefore, a well-designed classifier for the search results is needed, that would help not only to filter noise out, but also to identify more relevant and useful syllabi.

This chapter presents our work regarding automatic recognition of syllabus pages through text classification to build a syllabus collection. Issues related to the selection of appropriate features as well as classifier model construction using both generative models (Naïve Bayes – NB [John & Langley, 1995; Kim, Han, Rim, & Myaeng, 2006]) and discriminative counterparts (Support Vector Machines – SVM [Boser, Guyon, & Vapnik, 1992]) are discussed. Our results show that SVM outperforms NB in recognizing true syllabi.

BACKGROUND

There has been recent interest in collecting and studying the syllabus genre. A small set of digital library course syllabi was manually collected and carefully analyzed, especially with respect to their reading lists, in order to define the digital library curriculum [Pomerantz, Oh, Yang, Fox, & Wildemuth, 2006]. In the MIT OpenCourseWare project, 1,400 MIT course syllabi were manually collected and made publicly available, which required a lot of work by students and faculty.

Some efforts have already been devoted to automating the syllabus collection process. A syllabus acquisition approach similar to ours is described in [Matsunaga, Yamada, Ito, & Hirokaw, 2003]. However, their work differs from ours in the way syllabi are identified. They crawled Web pages from Japanese universities and sifted through them using a thesaurus with common words which occur often in syllabi. A decision tree was used to classify syllabus pages and entry pages (for example, a page containing links to all the syllabi of a particular course over time). Similarly, [Thompson, Smarr, Nguyen, & Manning, 2003] used a classification approach to classify education resources – especially syllabi, assignments, exams, and tutorials. Using the word features of each document, the authors were able to achieve very good performance (F_1 score: 0.98). However, this result is based upon their relative clean data set, only including the four kinds of education resources, which still took efforts to collect. We, on the other hand, to better apply to a variety of data domains, test and report our approach on search results for syllabi on the Web.

In addition, our genre feature selection work is also inspired by research on genre classification, which aims to classify data according to genre types by selecting features that distinguish one genre from another, i.e., identifying home pages in sets of web pages [Kennedy & Shepherd, 2005].

MAIN FOCUS

A text classification task usually can be accomplished by defining classes, selecting features, preparing a training corpus, and building a classifier. In order to build quickly an initial collection of CS syllabi, we obtained more than 8000 possible syllabus pages by programmatically searching using Google [Tungare et al., 2007]. After randomly examining the result set, we found it to contain many documents that were not truly syllabi: we refer to this as noise. To help with the task of properly identifying true syllabi, we defined true syllabi and false syllabi, and then selected features specific to the syllabus genre. We randomly sampled the collection to prepare a training corpus of size 1020. All 1020 files were in one of the following formats: HTML, PDF, PostScript, or Text. Finally, we applied Naïve Bayes, Support Vector Machines, and its variants to learn classifiers to produce the syllabus repository.

Class Definition

A syllabus component is one of the following: course code, title, class time, class location, offering institute, teaching staff, course description, objectives, web site, prerequisite, textbook, grading policy, schedule, assignment, exam, or resource. A true syllabus is a page that describes a course by including most of these syllabus components, which can be located in the current page or be obtained by following outgoing links. A false syllabus (or noise) is a page for other purposes (such as an instructor's homepage with a link to syllabi for his/her teaching purpose) instead of describing a course.

The two class labels were assigned by three team members to the 1020 samples with unanimous agreement. A skewed class distribution was observed in the sample set with 707 true syllabus and 313 false syllabus pages. We used this sample set as our training corpus.

Feature Selection

In a text classification task, a document is represented as a vector of features usually from a high dimensional space that consists of unique words occurring in documents. A good feature selection method reduces the feature space so that most learning algorithms can handle and contribute to high classification accuracy. We applied three feature selection methods in our study: general feature selection, genre-specific feature selection, and a hybrid of the two.

1. *General Features* - In a study of feature selection methods for text categorization tasks [Yang & Pedersen, 1997], the authors concluded that Document Frequency (DF) is a good choice since

its performance was similar to the one deemed best such as Information Gain and Chi Square, and it is simple and efficient. Therefore, we chose DF as our general feature selection method. In our previous work [Yu et al., 2008], we concluded that a DF threshold of 30 is a good setting to balance the computation complexity and classification accuracy. With such a feature selection setting, we obtained 1754 features from 63963 unique words in the training corpus.

2. *Genre Features* - Each defined class has its own characteristics other than general features. Many keywords such as 'grading policy' occur in a true syllabus probably along with a link to the content page. On the other hand, a false syllabus might contain syllabus keyword without enough keywords related to the syllabus components. In addition, the position of a keyword within a page matters. For example, a keyword within the anchor text of a link or around the link would suggest a syllabus component outside the current page. A capitalized keyword at the beginning of a page would suggest a syllabus component with a heading in the page. Motivated by the above observations, we manually selected 84 features to classify our data set into the four classes. We used both content and structure features for syllabus classification, as they have been found useful in the detection of other genres [Kennedy & Shepherd, 2005]. These features mainly concern the occurrence of keywords, the positions of keywords, and the co-occurrence of keywords and links. Details of these features are in [Yu et al., 2008].

After extracting free text from these documents, our training corpus consisted of 63963 unique terms, We represented it by the three kinds of feature attributes: 1754 unique general features, 84 unique genre features, and 1838 unique features in total. Each of these feature attributes has a numeric value between 0.0 and 1.0.

Classifiers

NB and SVM are two well-known best performing supervised learning models in text classification applications [Kim, Han, Rim, & Myaeng, 2006; Joachims, 1998]. NB, a simple and efficient approach, succeeds in various data mining tasks, while SVM, a highly complex one, outperforms NB especially in text mining tasks [Kim, Han, Rim, & Myaeng, 2006]. We describe them below.

1. *Naïve Bayes* - Naïve Bayes classifier can be viewed as a Bayesian network where feature attributes X_1, X_2, …, X_n are conditionally independent given the class attribute C [John & Langley, 1995]. Let C be a random variable and X be a vector of random variables X_1, X_2, …, X_n. The probability of a document *x* being in class *c* is calculated using Bayes' rule as below. The document will be classified into the most probable class.

$$p(C = c \mid X = x) = \frac{p(X = x \mid C = c)\, p(C = c)}{p(X = x)}$$

Since feature attributes (x_1, x_2, ..., x_n) represent the document *x,* and they are assumed to be conditionally independent, we can obtain the equation below.

$$p(X = x \mid C = c) = \prod_i p(X_i = x_i \mid C = c)$$

An assumption to estimate the above probabilities for numeric attributes is that the value of such an attribute follows a normal distribution within a class. Therefore, we can estimate $p(X_i = x_i \mid C = c)$ by using the mean and the standard deviation of such a normal distribution from the training data.

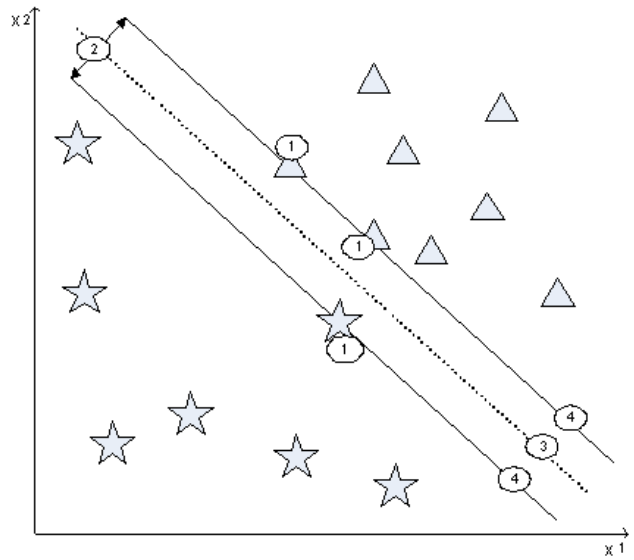
Such an assumption for the distribution may not hold for some domains. Therefore, we also applied the kernel method from [John & Langley, 1995] to estimate the distribution of each numeric attribute in our syllabus classification application.

2. *Support Vector Machines* - It is a two-class classifier (Figure 1) that finds the hyperplane maximizing the minimum distance between the hyperplane and training data points [Boser, Guyon, & Vapnik, 1992]. Specifically, the hyperplane $\omega^T x + \gamma$ is found by minimizing the objective function:

$$\frac{1}{2} \| \omega \|^2 \text{ such that } D(A\omega - e\gamma) \geq e$$

The margin is

Figure 1. Support Vector Machines where the hyperplane (3) is found to separate two classes of objects (represented here by stars and triangles, respectively) by considering the margin (i.e., distance) (2) of two support hyperplanes (4) defined by support vectors (1). An special case is depicted here that each object only has two feature variables x1 and x2.



$$\frac{2}{\|\omega\|^2}.$$

D is a vector of classes of training data, i.e., each item in *D* is +1 or −1. *A* is the matrix of feature values of training data. *e* is the vector of ones. After ω and γ are estimated from training data, a testing item *x* will be classified as +1 if

$\omega^T x + \gamma > 0$ and −1 otherwise.

The soft margin hyperplane [Cortes & Vapnik, 1995] was proposed to allow for the case where the training data points cannot be split without errors. The modified objective function is

$$\frac{1}{2}\|\omega\|^2 + \sum_i \varepsilon_i \text{ such that } D(A\omega - e\gamma) \geq e - \xi$$

where $\xi = (\varepsilon_1 \ldots \varepsilon_n)^T$ and ε_i measures the degree of misclassification of the *i*th training data point during the training. It considers minimizing the errors while maximizing the margins.

In some cases, it is not easy to find the hyperplane in the original data space, in which case the original data space has to be transformed into a higher dimensional space by applying kernels [Boser, Guyon, & Vapnik, 1992]. Common used kernels include polynomial, radial basis function, Gaussian radial basis function, and sigmoid. In our comparative study, we only tested SVM with a polynomial kernel. In addition, sequential minimal optimization (SMO) [Platt, 1999], a fast nonlinear optimization method, was employed during the training process to accelerate training.

Evaluation Results and Discussions

Evaluation Setups - We applied the classification models discussed above (five settings in total implemented with the Weka package [Witten & Frank, 2005]) on the training corpus with the three different feature sets. In the rest of this paper, we refer to the SVM implemented using the SMO simply as *'SMO'* for short; the one with the polynomial kernel as *'SMO-K'*, Naïve Bayes with numeric features estimated by Gaussian distribution as *'NB'*, and the one with kernel as *'NB-K'*. We used tenfold cross validation to estimate the classification performance as measured by F_1. Tenfold cross validation estimates the average classification performance by splitting a training corpus into ten parts and averaging the performance in ten runs, each run with nine of these as a training set and the rest as a testing set. F_1 is a measure that trades off precision and recall. It provides an overall measure of classification performance. For

each class, the definitions of the measures are as follows. A higher F_1 value indicates better classification performance.

- Precision is the percentage of the correctly classified positive examples among all the examples classified as positive.
- Recall is the percentage of the correctly classified positive examples among all the positive examples.
- $$F_1 = \frac{2 \times \Pr ecision \times \mathrm{Re} call}{\Pr ecision + \mathrm{Re} call}$$

(2) *Findings and Discussions* – The following are the main four findings from our experiments.

First, SVM outperforms NB in syllabus classification in the average case (Figure 2). On average, SMO performed best at the F_1 score of 0.87, 15% better than NB in terms of the true syllabus class and 1% better in terms of the false syllabus class. The best setting for our task is SMO with the genre feature selection method, which achieved an F_1 score of 0.88 in recognizing true syllabi and 0.71 in recognizing false syllabi.

Second, the kernel settings we tried in the experiments were not helpful in the syllabus classification task. Figure 2 indicates that SMO with kernel settings perform rather worse than that without kernels.

Third, the performance with genre features settings outperforms those with general features settings and hybrid feature settings. Figure 3 shows this performance pattern in the SMO classifier setting; other classifiers show the same pattern. We also found that the performance with hybrid features settings is dominated by the general features among them. It is probably because the number of the genre features is very small, compared to the number of general features. Therefore, it might be useful to test new ways of mixing genre features and general features to take advantage of both of them more effectively.

Finally, at all settings, better performance is achieved in recognizing true syllabi than in recognizing false syllabi. We analyzed the classification results with the best setting and found that 94 of 313 false syllabi were classified as true ones mistakenly. It is likely that the skewed distribution in the two classes makes classifiers favor true syllabus class given an error-prone data point. Since we probably provide no appropriate information if we misclassify a true syllabus as a false one, our better performance in the true syllabus class is satisfactory.

FUTURE WORK

Although general search engines such as Google meet people's basic information needs, there are still possibilities for improvement, especially with genre-specific search. Our work on the syllabus genre successfully indicates that machine learning techniques can contribute to genre-specific search and classification. In the future, we plan to improve the classification accuracy from multiple perspectives such as defining

Figure 2. Classification performance of different classifiers on different classes measured in terms of F_1

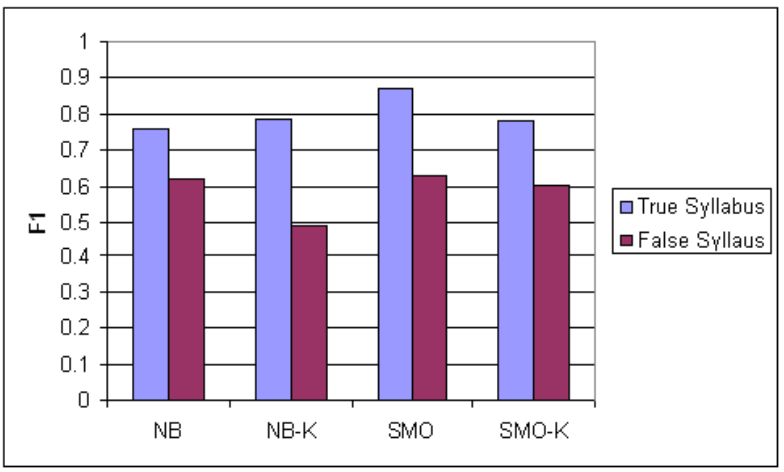
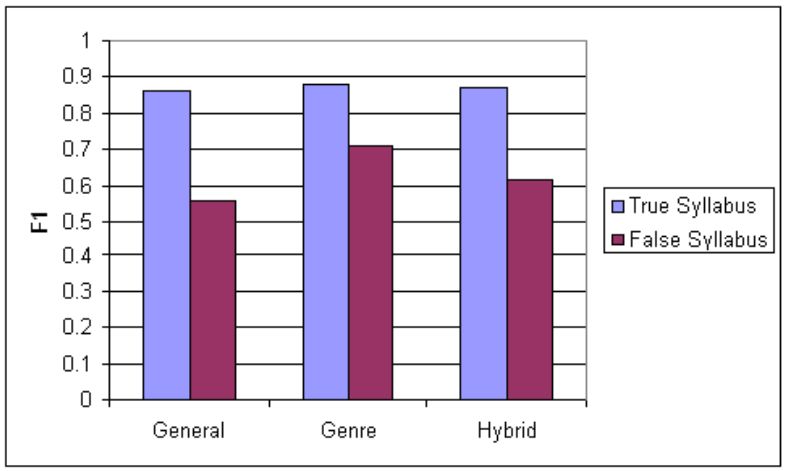


Figure 3. Classification performance of SMO on different classes using different feature selection methods measured in terms of F_1



more genre-specific features and applying many more state-of-the-art classification models.

Our work on automatically identifying syllabi among a variety of publicly available documents will also help build a large-scale educational resource repository for the academic community. We are obtaining more syllabi to grow our current repository by manual submissions to our repository website, http://syllabus.cs.vt.edu, from people who would like to share their educational resources with one another.

CONCLUSION

In this chapter, we presented our work on automatically identifying syllabi from search results on the Web. We proposed features specific to the syllabus genre and compared them with general features obtained by the document frequency method. Our results showed the promising future of genre-specific feature selection methods regarding the computation complexity and improvement space. We also employed state-of-the-art machine learning techniques for automatic classification. Our results indicated that support vector machines were a good choice for our syllabus classification task.

REFERENCES

Baeza-Yates, R. A. & Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc.

Boser, B. E., Guyon, I. M. & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on computational learning theory* (pp. 144–152). New York, NY, USA: ACM Press.

Cortes, C. & Vapnik, V. (1995). Support-Vector Networks. *Machine Learning* 20, 3 (Sep. 1995), 273-297.

Fan, W., Wallace, L., Rich, S., & Zhang, Z. (2006). Tapping the power of text mining. *Communications of the ACM* 49, 9 (Sep. 2006), 76-82

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the European Conference on Machine Learning* (pp.137–142). Heidelberg, DE: Springer.

John, G. H., & Langley, P. (1995). Estimating continuous distributions in Bayesian classifiers. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence* (pp. 338–345).

Kennedy A. & Shepherd M. (2005). Automatic identification of home pages on the web. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences - Track 4*. Washington, DC, USA: IEEE Computer Society.

Kim, S.-B., Han, K.-S., Rim, H.-C., & Myaeng, S. H. (2006). Some effective techniques for naive bayes text classification. *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 11, 1457–1466.

Matsunaga, Y., Yamada, S., Ito, E., & Hirokaw S. (2003) A web syllabus crawler and its efficiency evaluation. In *Proceedings of International Symposium on Information Science and Electrical Engineering* (pp. 565-568).

Pomerantz, J., Oh, S., Yang, S., Fox, E. A., & Wildemuth, B. M. (2006) The core: Digital library education in library and information science programs. *D-Lib Magazine*, vol. 12, no. 11.

Platt, J. C. (1999). Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C. J. Burges, & A. J. Smola, (Eds.) *Advances in Kernel Methods: Support Vector Learning* (pp. 185-208). MIT Press, Cambridge, MA.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys.* 34, 1 (Mar. 2002), 1-47.

Thompson, C. A., Smarr, J., Nguyen, H. & Manning, C. (2003) Finding educational resources on the web: Exploiting automatic extraction of metadata. In *Proceedings of European Conference on Machine Learning Workshop on Adaptive Text Extraction and Mining*.

Tungare, M., Yu, X., Cameron, W., Teng, G., Pérez-Quiñones, M., Fox, E., Fan, W., & Cassel, L. (2007). Towards a syllabus repository for computer science courses. In *Proceedings of the 38th Technical Symposium on Computer Science Education* (pp. 55-59). SIGCSE Bull. 39, 1.

Witten, I. H., & Frank E. (2005). Data Mining: Practical Machine Learning Tools and Techniques (Second Edition). Morgan Kaufmann.

Yang Y. & Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning* (pp. 412–420). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Yu, X., Tungare, M., Fan, W., Pérez-Quiñones, M., Fox, E. A., Cameron, W., Teng, G., & Cassel, L. (2007). Automatic syllabus classification. In *Proceedings of the Seventh ACM/IEEE-CS Joint Conference on Digital Libraries* (pp. 440-441). New York, NY, USA: ACM Press.

Yu, X., Tungare, M., Fan, W., Yuan, Y., Pérez-Quiñones, M., Fox, E. A., Cameron, W., & Cassel, L. (2008). Automatic syllabus classification using support vector machines. (To appear in) M. Song & Y. Wu (Eds.) *Handbook of Research on Text and Web Mining Technologies*. Idea Group Inc.

KEY TERMS

False Syllabus: A page that does not describe a course.

Feature Selection: A method to reduce the high dimensionality of the feature space by selecting features that are more representative than others. In text classification, usually the feature space consists of unique terms occurring in the documents.

Genre: Information presented in a specific format, often with certain fields and subfields associated closely with the genre; e.g. syllabi, news reports, academic articles, etc.

Model Testing: A procedure performed after model training that applies the trained model to a different data set and evaluates the performance of the trained model.

Model Training: A procedure in supervised learning that generates a function to map inputs to desired outputs. In text classification, a function is generated to map a document represented by features into known classes.

Naïve Bayes (NB) Classifiers: A classifier modeled as a Bayesian network where feature attributes are conditionally independent of class attributes.

Support Vector Machines (SVM): A supervised machine learning approach used for classification and regression to find the hyperplane maximizing the minimum distance between the plane and the training data points.

Syllabus Component: One of the following pieces of information: course code, title, class time, class location, offering institute, teaching staff, course description,

objectives, web site, prerequisite, textbook, grading policy, schedule, assignment, exam, or resource.

Text Classification: The problem of automatically assigning predefined classes to text documents.

True Syllabus: A page that describes a course; it includes many of the syllabus components described above, which can be located in the current page or be obtained by following outgoing links.

Automatic Music Timbre Indexing

Xin Zhang
University of North Carolina at Charlotte, USA

Zbigniew W. Ras
University of North Carolina, Charlotte, USA

INTRODUCTION

Music information indexing based on timbre helps users to get relevant musical data in large digital music databases. Timbre is a quality of sound that distinguishes one music instrument from another among a wide variety of instrument families and individual categories. The real use of timbre-based grouping of music is very nicely discussed in (Bregman, 1990).

Typically, an uncompressed digital music recording, in form of a binary file, contains a header and a body. A header stores file information such as length, number of channels, rate of sample frequency, etc. Unless being manually labeled, a digital audio recording has no description on timbre, pitch or other perceptual properties. Also, it is a highly nontrivial task to label those perceptual properties for every piece of music object based on its data content. Lots of researchers have explored numerous computational methods to identify the timbre property of a sound. However, the body of a digital audio recording contains an enormous amount of integers in a time-order sequence. For example, at a sample frequency rate of 44,100Hz, a digital recording has 44,100 integers per second, which means, in a one-minute long digital recording, the total number of the integers in the time-order sequence will be 2,646,000, which makes it a very big data item. Being not in form of a record, this type of data is not suitable for most traditional data mining algorithms.

Recently, numerous features have been explored to represent the properties of a digital musical object based on acoustical expertise. However, timbre description is basically subjective and vague, and only some subjective features have well defined objective counterparts, like brightness, calculated as gravity center of the spectrum. Explicit formulation of rules of objective specification of timbre in terms of digital descriptors will formally express subjective and informal sound characteristics. It is especially important in the light of human perception of sound timbre. Time-variant information is necessary for correct classification of musical instrument sounds because quasi-steady state, where the sound vibration is stable, is not sufficient for human experts. Therefore, evolution of sound features in time should be reflected in sound description as well. The discovered temporal patterns may better express sound features than static features, especially that classic features can be very similar for sounds representing the same family or pitch, whereas changeability of features with pitch for the same instrument makes sounds of one instrument dissimilar. Therefore, classical sound features can make correct identification of musical instruments independently on the pitch very difficult and erroneous.

BACKGROUND

Automatic content extraction is clearly needed and it relates to the ability of identifying the segments of audio in which particular predominant instruments were playing. Instruments having rich timbre are known to produce overtones, which result in a sound with a group of frequencies in clear mathematical relationships (so-called harmonics). Most western instruments produce harmonic sounds. Generally, identification of musical information can be performed for audio samples taken from real recordings, representing waveform, and for MIDI (Musical Instrument Digital Interface) data. MIDI files give access to highly structured data. So, research on MIDI data may basically concentrate on higher level of musical structure, like key or metrical information. Identifying the predominant instruments, which are playing in the multimedia segments, is

even more difficult. Defined by ANSI as the attribute of auditory sensation, timbre is rather subjective: a quality of sound, by which a listener can judge that two sounds, similarly presented and having the same loudness and pitch, are different. Such definition is subjective and not of much use for automatic sound timbre classification. Therefore, musical sounds must be very carefully parameterized to allow automatic timbre recognition. There are a number of different approaches to sound timbre (Balzano, 1986; Cadoz, 1985). Dimensional approach to timbre description was proposed by (Bregman, 1990). Sets of acoustical features have been successfully developed for timbre estimation in monophonic sounds where mono instruments were playing. However, none of those features can be successfully applied to polyphonic sounds, where two or more instruments were playing at the same time, since those features represent the overlapping sound harmonics as a whole instead of individual sound sources.

This has brought the research interest into Blind Source Separation (BBS) and independent component analysis (ICA) for musical data. BBS is to estimate original sound sources based on signal observations without any knowledge on the mixing and filter procedure. ICA is to separate sounds by linear models of matrix factorization based on the assumption that each sound source is statistically independent. Based on the fact that harmonic components have significant energy, harmonics tracking together with Q-Constant Transform and Short Time Fourier Transform have been applied to sound separation (Dziubinski, Dalka and Kostek 2005; Herrera, Peeters and Dubnov 2003; Zhang and Ras 2006B). The main steps in those researches include processing polyphonic sounds into monophonic sounds, extracting features from the resultant monophonic sounds, and then performing classification.

MAIN FOCUS

Current research in timbre recognition for polyphonic sounds can be summarized into three steps: sound separation, feature extraction and classification. Sound separation has been used to process polyphonic sounds into monophonic sounds by isolating sound sources; features have been used to represent the sound behaviors in different domains; then, classification shall be performed based on the feature values by various classifiers.

Sound Separation

In a polyphonic sound with multiple pitches, multiple sets of harmonics from different instrument sources are overlapping with each other. For example, in a sound mix where a sound in 3A of clarinet and a sound in 4C of violin were played at the same time, there are two sets of harmonics: one set is distributed near several integer multiples of 440Hz; the other spreads around integer multiples of 523.25Hz. Thus, the j^{th} harmonic peak of the k^{th} instrument can be estimated by searching a local peak in the vicinity of an integer multiple of the fundamental frequency. Consequently, k predominant instruments will result in k sets of harmonic peaks. Then, we can merge the resultant sets of harmonic peaks together to form a sequence of peaks H_p^j in an ascending order by the frequency, where three possible situations should be taken into consideration for each pair of neighbor peaks: the two immediate peak neighbors are from the same sound source; the two immediate peak neighbors are from two different sound sources; part of one of the peak and the other peak are from the same sound source. The third case is due to two overlapping peaks, where the frequency is the multiplication of the fundamental frequencies of two different sound sources. In this scenario, the system first partitions the energy between the two sound sources according to the ratio of the previous harmonic peaks of those two sound sources. Therefore, only the heterogeneous peaks should be partitioned. A clustering algorithm has been used for separation of energy between two immediate heterogeneous neighbor peaks. Considering the wide range of the magnitude of harmonic peaks, we may apply a coefficient to linearly scale each pair of immediate neighbor harmonic peaks to a virtual position along the frequency axis by a ratio of the magnitude values of the two harmonic peaks. Then the magnitude of each point between the two peaks is proportionally computed in each peak. For fast computation, a threshold for the magnitude of each FFT point has been applied, where only points with significant energy had been computed by the above formulas. We assume that a musical instrument is not predominant only when its total harmonic energy is significantly smaller than the average of the total harmonic energy of all sound sources. After clustering the energy, each FFT point in the analysis window has been assigned k coefficients, for each predominant instrument accordingly.

Feature Extraction

Methods in research on automatic musical instrument sound classification go back to last few years. So far, there is no standard parameterization used as a classification basis. The sound descriptors used are based on various methods of analysis in time domain, spectrum domain, time-frequency domain and cepstrum with Fourier Transform for spectral analysis being most common, such as Fast Fourier Transform, Short-time Fourier Transform, Discrete Fourier Transform, and so on. Also, wavelet analysis gains increasing interest for sound and especially for musical sound analysis and representation. Based on recent research performed in this area, MPEG proposed an MPEG-7 standard, in which it described a set of low-level sound temporal and spectral features. However, a sound segment of note played by a music instrument is known to have at least three states: transient state, quasi-steady state and decay state. Vibration pattern in a transient state is known to significantly differ from the one in a quasi-steady state. Temporal features in differentiated states enable accurate instrument estimation.

These acoustic features can be categorized into two types in terms of size:

- **Acoustical instantaneous features in time series:** A huge matrix or vector, where data in each row describe a frame, such as Power Spectrum Flatness, and Harmonic Peaks, etc. The huge size of data in time series is not suitable for current classification algorithms and data mining approaches.
- **Statistical summation of those acoustical features:** A small vector or single value, upon which classical classifiers and analysis approaches can be applied, such as Tristimulus (Pollard and Jansson, 1982), Even/Odd Harmonics (Kostek and Wieczorkowska, 1997), averaged harmonic parameters in differentiated time domain (Zhang and Ras, 2006A), etc.

Machine Learning Classifiers

The classifiers, applied to the investigations on musical instrument recognition and speech recognition, represent practically all known methods: Bayesian Networks (Zweig, 1998; Livescu and Bilmes, 2003), Decision Tree (Quinlan, 1993; Wieczorkowska, 1999), K-Nearest Neighbors algorithm (Fujinaga and McMillan 2000; Kaminskyj and Materka 1995), Locally Weighted Regression (Atkeson and Moore, 1997), Logistic Regression Model (le Cessie and Houwelingen, 1992), Neural Networks (Dziubinski, Dalka and Kostek 2005) and Hidden Markov Model (Gillet and Richard 2005), etc. Also, hierarchical classification structures have been widely used by researchers in this area (Martin and Kim, 1998; Eronen and Klapuri, 2000), where sounds have been first categorized to different instrument families (e.g. the String Family, the Woodwind Family, the Percussion Family, etc), and then been classified into individual categories (e.g. Violin, Cello, Flute, etc.)

FUTURE TRENDS

The classification performance relies on sound items of the training dataset and the multi-pitch detection algorithms. More new temporal features in time-variation against background noise and resonance need to be investigated. Timbre detection of sounds with overlapping in homogeneous pitches from different instruments can be a very interesting and challenging area.

CONCLUSION

Timbre detection is one of the most important sub-tasks for content based indexing. In Automatic Music Timbre Indexing, timbre is estimated based on computation of the content of audio data in terms of acoustical features by machine learning classifiers. An automatic music timbre indexing system should have at least the following components: sound separation, feature extraction, and hierarchical timbre classification. We observed that sound separation based on multi-pitch trajectory significantly isolated heterogeneous harmonic sound sources in different pitches. Carefully designed temporal parameters in the differentiated time-frequency domain together with the MPEG-7 low-level descriptors have been used to briefly represent subtle sound behaviors within the entire pitch range of a group of western orchestral instruments. The results of our study also showed that Bayesian Network had a significant better performance than Decision Tree, Locally Weighted Regression and Logistic Regression Model.

REFERENCES

Atkeson, C.G., Moore A.W., and Schaal, S. (1997). Locally Weighted Learning for Control, *Artificial Intelligence Review*. Feb. 11(1-5), 75-113.

Balzano, G.J. (1986). What are Musical Pitch and Timbre? *Music Perception - an Interdisciplinary Journal*. 3, 297-314.

Bregman, A.S. (1990). Auditory Scene Analysis, the Perceptual Organization of Sound, *MIT Press*

Cadoz, C. (1985). Timbre et causalite, Unpublished paper, *Seminar on Timbre*, Institute de Recherche et Coordination Acoustique / Musique, Paris, France, April 13-17.

Dziubinski, M., Dalka, P. and Kostek, B. (2005) Estimation of Musical Sound Separation Algorithm Effectiveness Employing Neural Networks, *Journal of Intelligent Information Systems*, 24(2/3), 133–158.

Eronen, A. and Klapuri, A. (2000). Musical Instrument Recognition Using Cepstral Coefficients and Temporal Features. In proceeding of *the IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP*, Plymouth, MA, 753-756.

Fujinaga, I., McMillan, K. (2000) Real time Recognition of Orchestral Instruments, *International Computer Music Conference*, 141-143.

Gillet, O. and Richard, G. (2005) Drum Loops Retrieval from Spoken Queries, *Journal of Intelligent Information Systems*, 24(2/3), 159-177

Herrera. P., Peeters, G., Dubnov, S. (2003) Automatic Classification of Musical Instrument Sounds, *Journal of New Music Research*, 32(19), 3–21.

Kaminskyj, I., Materka, A. (1995) Automatic Source Identification of Monophonic Musical Instrument Sounds, *the IEEE International Conference On Neural Networks*, Perth, WA, 1, 189-194

Kostek, B. and Wieczorkowska, A. (1997). Parametric Representation of Musical Sounds, *Archive of Acoustics*, Institute of Fundamental Technological Research, Warsaw, Poland, 22(1), 3-26.

le Cessie, S. and van Houwelingen, J.C. (1992). Ridge Estimators in Logistic Regression, *Applied Statistics*, 41, (1), 191-201.

Livescu, K., Glass, J., and Bilmes, J. (2003). Hidden Feature Models for Speech Recognition Using Dynamic Bayesian Networks, in Proc. *Euro-speech*, Geneva, Switzerland, September, 2529-2532.

Martin, K.D., and Kim, Y.E. (1998). Musical Instrument Identification: A Pattern-Recognition Approach, in *the 136th Meeting of the Acoustical Society of America*, Norfolk, VA.

Pollard, H.F. and Jansson, E.V. (1982). A Tristimulus Method for the Specification of Musical Timbre. *Acustica*, 51, 162-171

Quinlan, J.R. (1993). *C4.5: Programs for Machine Learning*, Morgan Kaufmann, San Mateo, CA.

Wieczorkowska, A. (1999). Classification of Musical Instrument Sounds using Decision Trees, in *the 8th International Symposium on Sound Engineering and Mastering*, ISSEM'99, 225-230.

Wieczorkowska, A., Wroblewski, J., Synak, P., and Slezak, D. (2003). Application of Temporal Descriptors to Musical Instrument Sound, Journal of Intelligent Information Systems, *Integrating Artificial Intelligence and Database Technologies*, July, 21(1), 71-93.

Zhang, X. and Ras, Z.W. (2006A). Differentiated Harmonic Feature Analysis on Music Information Retrieval For Instrument Recognition, *proceeding of IEEE International Conference on Granular Computing*, May 10-12, Atlanta, Georgia, 578-581.

Zhang, X. and Ras, Z.W. (2006B). Sound Isolation by Harmonic Peak Partition for Music Instrument Recognition, *Special Issue on Knowledge Discovery,* (Z. Ras, A. Dardzinska, Eds)*, in Fundamenta Informaticae Journal*, IOS Press, 2007, will appear

Zweig, G. (1998). Speech Recognition with Dynamic Bayesian Networks, Ph.D. dissertation, Univ. of California, Berkeley, California.

ISO/IEC, JTC1/SC29/WG11. (2002). MPEG-7 Overview. Available at http://mpeg.telecomitalialab.com/standards/mpeg-7/mpeg-7.htm

KEY TERMS

Automatic Indexing: Automatically identifies precise, relevant clips of content within audio sources.

Feature Extraction: The process of generating a set of descriptors or characteristic attributes from a binary musical file.

Harmonic: A set of component pitches in mathematical relationship with the fundamental frequency.

Hierarchical Classification: Classification in a top-down order. First identify musical instrument family types, and then categorize individual or groups of instruments within the instrument family.

Machine Learning: A study of computer algorithms that improve their performance automatically based on previous results.

MPEG-7: A Multimedia Content Description Interface standardizes descriptions for audio-visual content by Moving Picture Experts Group.

Quasi-Steady State: A steady state where frequencies are in periodical patterns.

Short-Time Fourier Transform: By using an analysis window, e.g., a hamming window, signal is evaluated with elementary functions that are localized in time and frequency domains simultaneously.

Sound Separation: The process of isolating sound sources within a piece of sound.

Timbre: Describes those characteristics of sound, which allow the ear to distinguish one instrument from another.

Time-Frequency Domain: A time series of analysis windows, where patterns are described in frequency domain.

A Bayesian Based Machine Learning Application to Task Analysis

Shu-Chiang Lin
Purdue University, USA

Mark R. Lehto
Purdue University, USA

INTRODUCTION

Many task analysis techniques and methods have been developed over the past decades, but identifying and decomposing a user's task into small task components remains a difficult, impractically time-consuming, and expensive process that involves extensive manual effort (Sheridan, 1997; Liu, 1997; Gramopadhye and Thaker, 1999; Annett and Stanton, 2000; Bridger, 2003; Stammers and Shephard, 2005; Hollnagel, 2006; Luczak et al., 2006; Morgeson et al., 2006). A practical need exists for developing automated task analysis techniques to help practitioners perform task analysis efficiently and effectively (Lin, 2007). This chapter summarizes a Bayesian methodology for task analysis tool to help identify and predict the agents' subtasks from the call center's naturalistic decision making's environment.

BACKGROUND

Numerous computer-based task analysis techniques have been developed over the years (Gael, 1988; Kirwan and Ainsworth, 1992; Wickens and Hollands, 2000; Hollnagel, 2003; Stephanidis and Jacko, 2003; Diaper and Stanton, 2004; Wilson and Corlett, 2005; Salvendy, 2006; Lehto and Buck, 2008). These approaches are similar in many ways to methods of knowledge acquisition commonly used during the development of expert systems (Vicente, 1999; Schraagen et al., 2000; Elm et al., 2003; Shadbolt and Burton, 2005). Several taxonomies exist to classify knowledge elicitation approaches. For example, Lehto et al. (1992) organize knowledge elicitation methods (including 140 computer-based tools), identified in an extensive review of 478 articles, into three categories: manual methods, interactive or semi-automated methods, and automated or machine learning methods. Manual methods such as protocol analysis or knowledge organization are especially useful as an initial approach because they can be used to effectively retrieve structure and formalize knowledge components, resulting in a knowledge base that is accurate and complete (Fujihara, et al., 1997). Studies such as Trafton et al. (2000) have shown this technique can capture the essence of qualitative mental models used in complex visualization and other tasks. The drawbacks of this technique are similar to those of classic task analysis techniques in that they involve extensive manual effort and may interfere with the expert's ability to perform the task. Semi-automated methods generally utilize computer programs to simplify applications of the manual methods of knowledge acquisition. The neural network model is one of the methods in common use today, especially when learning and recognition of patterns are essential (Bhagat, 2005). A neural network can self-update its processes to provide better estimates and results with further training. However, one arguable disadvantage is that this approach may require considerable computational power should the problem be somewhat complex (Dewdney, 1997).

Automated methods or machine learning based methods primarily focus on learning from recorded data rather than through direct acquisition of knowledge from human experts. Many variations of commonly used machine learning algorithms can be found in the literature. In general, the latter approach learns from examples-guided deductive/inductive processes to infer rules applicable to other similar situations (Shalin, et al., 1988; Jagielska et al., 1999; Wong & Wang, 2003; Alpaydın, 2004; Huang et al., 2006; Bishop, 2007).

MAIN FOCUS

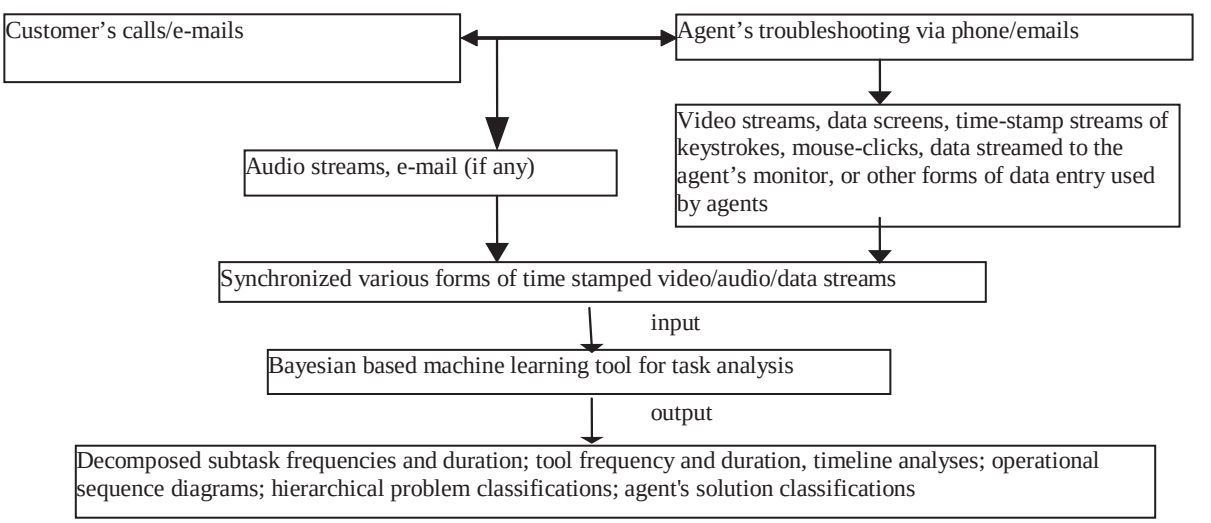
The Bayesian framework provides a potentially more applicable method of task analysis compared to competing approaches such as neural networks, natural language processing methods, or linguistic models. Two Bayesian methods are often proposed: naïve Bayes and fuzzy Bayes. Over the decades, studies such as those of Bookstein, (1985), Evans and Karwowski (1987), Lehto and Sorock (1996), Chatterjee (1998), Yamamoto and Sagisaka (1999), Zhu and Lehto (1999), Qiu and Agogino (2001), Hatakeyama et al. (2003), Zhou and Huang (2003), Leman and Lehto (2003), Wellman et al. (2004), and Bolstad (2004) have shown that statistical machine learning within the framework of fuzzy Bayes can be more efficient when the assumptions of independence are violated. McCarthy (2002) found that fuzzy Bayes gave the highest success rate for print defect classification compared to ID3, C4.5, and individual keyword comparison algorithms. Noorinaeini and Lehto (2007) compare the accuracy of three Singular Value Decomposition (SVD) based Bayesian/Regression models and conclude that all three models are capable of learning from human experts to accurately categorize cause-of-injury codes from injury narrative.

Case studies have contributed to both theoretical and empirical research in the naturalistic decision making environment (Zsambok, 1997; Klein, 1998; Todd & Gigerenzer, 2001; Hutton et al, 2003). The following discussion presents a brief case study illustrating the application of a Bayesian method to task analysis. This particular study here focuses on describing what takes place in a call center, when the customer calls to report various problems and the knowledge agent helps troubleshoot remotely. In this example, the conversation between agent and customer was recorded and manipulated to form a knowledge database as input to the Bayesian based machine learning tool.

Model Development

Figure 1 illustrates important elements of the dialog between a call center knowledge agent and customer. The arrows indicate data flow. The dialog between the customer and the knowledge agent can be recorded using several methods. For example, if the customer uses e-mail, these conversations are directly available in written form. The knowledge agent's troubleshooting processes similarly could be recorded in video streams, data screens, time-stamp streams of keystrokes, mouse-clicks, data streamed to the agent's monitor, or various forms of data entry used by agents. These data streams can be synchronized with a time-stamp as input for the Bayesian based machine learning tool.

Figure 1, Model of Bayesian based machine learning tool for task analysis[1]



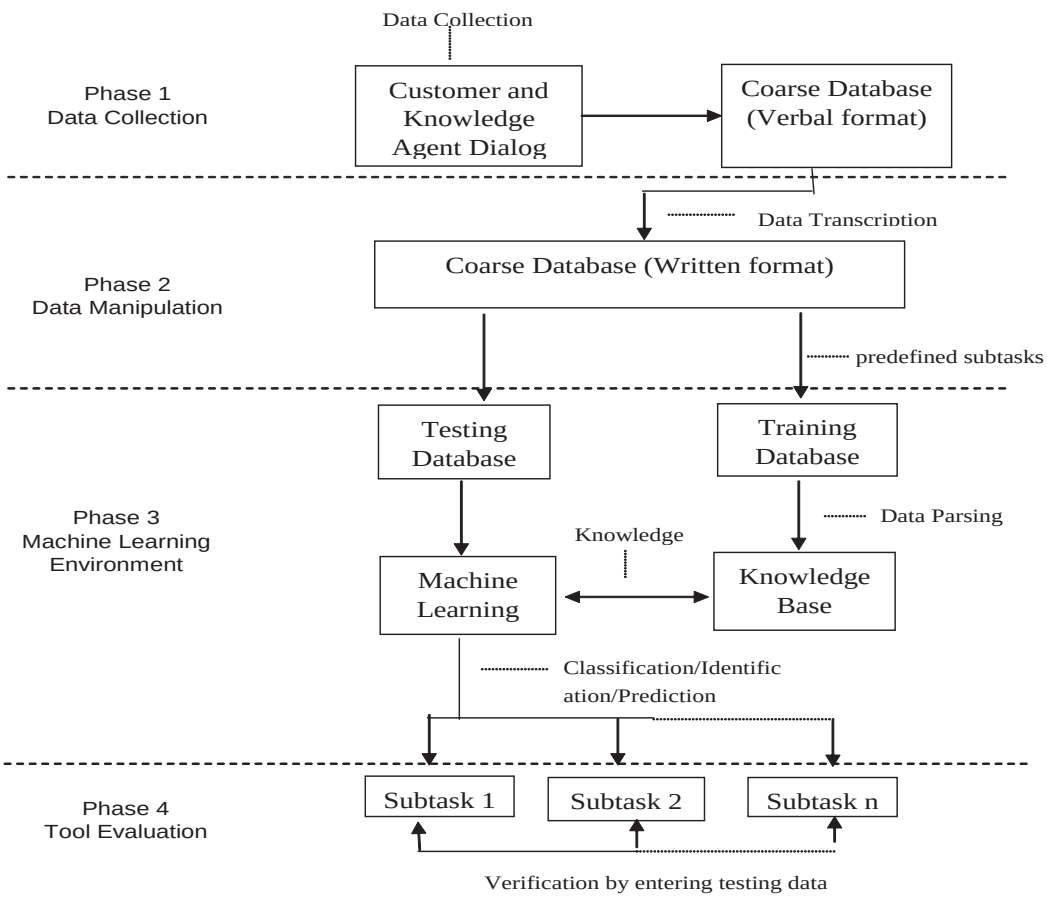
The anticipated output of the tool could be a set of decomposed subtasks/tools with associated frequencies and duration, timeline analyses, operational sequence diagrams, product hierarchical problem classifications such as software issues, hardware issues, network issues, media issues, or print quality issues, and agent solution classifications such as dispatching an on-site technician, sending out parts, or escalating the problem to the attention of product engineers. Potential benefits of this approach might include more efficient and effective integration of the human element into system operations, better allocation of task functions, human-computer interface redesigns, and revisions of agent training materials.

Methodology

As illustrated in Figure 2, the process followed to implement this methodology consists of four phases:

1. Recorded phone conversations between the customer and the knowledge agent are transcribed into a written format. Preferably this data is collected from several knowledge agents using a non-intrusive process. Following such an approach in the field will produce a large amount of realistic and naturalistic case-based information, which is unlikely to be obtained through lab-controlled methodologies that require making assumptions

Figure 2, Four phases to the development of Bayesian based machine learning tool[2]



that have arguably little face validity. Such as approach also helps balance out the effect of many biasing factors such as individual differences among the knowledge agents, unusual work activities followed by a particular knowledge agent, or sampling problems related to collecting data from only a few or particular groups of customers with specific questions.

2. The next step is to document the agents' troubleshooting process and define the subtask categories. The assigned subtasks can then be manually assigned into either the training set or testing set. Development of subtask definitions will normally require inputs from human experts. In addition to the use of transcribed verbal protocols, the analyst might consider using synchronized forms of time stamped video/audio/data streams.
3. In our particular study, the fuzzy Bayes model was developed (Lin, 2006) during the text mining process to describe the relation between the verbal protocol data and assigned subtasks. For fuzzy Bayes method, the expression below is used to classify subtasks into categories:

$$P(S_i \mid E) = MAX_j \frac{P(E_j \mid S_i)P(S_i)}{P(E_j)}$$

where $P(S_i|E)$ is the posterior probability of subtask S_i is true given the evidence E (words used by the agent and the customer) is present,
$P(E_j|S_i)$ is the conditional probability of obtaining the evidence E_j given that the subtask S_i is true,
$P(S_i)$ is the prior probability of the subtask being true prior to obtaining the evidence E_j, and
"MAX" is used to assign the maximum value of calculated $P(E_j|S_i)*P(S_i)/P(E_j)$.
When agent performs subtask A_i, words used by the agent and the customer are expressed by word vectors

$WA_i = (WA_{i1}, WA_{i2}, \ldots, WA_{iq})$ and $WC_i = (WC_{i1}, WC_{i2}, \ldots, WC_{iq})$ respectively,

where $WA_{i1}, WA_{i2}, \ldots, WA_{iq}$ are the q words in the i^{th} agent's dialog/narrative;
$WC_{i1}, WC_{i2}, \ldots, WC_{iq}$ are the q words in the i^{th} customer's dialog/narrative.
A_i is considered potentially relevant to WA_i, WC_{i-1}, and WC_i for i greater than 1.

The posterior probability of subtask A_i is calculated as follows:

$P(A_i|WA_i,WC_i,WC_{i-1})=MAX[P(WA_i|A_i)*P(A_i)/P(WA_i), P(WC_i|A_i)*P(A_i)/P(WC_i), P(WC_{i-1}|A_i)*P(A_i)/P(WC_{i-1})]$
$=MAX[MAX_j[P(WA_{ij}|A_i)*P(A_i)/P(WA_{ij})], MAX_j[P(WC_{ij}|A_i)*P(A_i)/P(WC_{ij})], MAX_j[P(WC_{(i-1)j}|A_i)*P(A_i)/P(WC_{(i-1)j})]]$ for j=1,2,….q

To develop the model, keywords were first parsed from the training set to form a knowledge base. The Bayesian based machine learning tool then learned from the knowledge base. This involved determines combinations of words appearing in the narratives that could be candidates for subtask category predictors. These words were then used to predict subtask categories, which was the output of the fuzzy Bayes model.

4. The fuzzy Bayes model was tested on the test set and the model performance was evaluated in terms of hit rate, false alarm rate, and sensitivity value. The model training and testing processes were repeated ten times to allow cross-validation of the accuracy of the predicted results. The testing results showed that the average hit rate (56.55%) was significantly greater than the average false alarm rate (0.64%), and a sensitivity value of 2.65 greater than zero.

FUTURE TRENDS

The testing results reported above suggest that the fuzzy Bayesian based model is able to learn and accurately predict subtask categories from the telephone conversation between the customers and the knowledge agents. These results are encouraging given the complexity of the tasks addressed. That is, the problem domain included 24 different agents, 55 printer models, 75 companies, 110 customers, and over 70 technical issues. Future studies are needed to further evaluate model performance that includes topics such as alternative groupings of subtasks and words, as well as use of word sequences. Other research opportunities include further development and exploration of a variety of Bayesian models, as well as comparison of model

performance to classification algorithms such as ID3 and C4.5. Researchers also might explore implementation of the Bayesian based model to other service industries. For example, in health care applications, a similar tool might be used to analyze tasks performed by clerks or nursing staff.

CONCLUSION

Bayesian based machine learning methods can be combined with classic task analysis methods to help practitioners analyze tasks. Preliminary results indicate this approach successfully learned how to predict subtasks from the telephone conversations between customers and call center agents. These results support the conclusion that Bayesian methods can serve as a practical methodology in the field of important research area of task analysis as well as other areas of naturalistic decision making.

REFERENCES

Alpaydın, E. (2004). Introduction to Machine Learning (Adaptive Computation and Machine Learning). *MIT Press*. Cambridge, MA.

Annett, J. and Stanton, N.A. (Eds.) (2000). *Task Analysis*. London: Taylor and Francis.

Bhagat, P.M. (2005). *Pattern Recognition in Industry*. Elsevier. Amsterdam, The Netherlands.

Bishop, C. M. (2007). *Pattern Recognition and Machine Learning*. Springer.

Bolstad, W. M. (2004). *Introduction to Bayesian Statistics*. John Wiley

Bookstein, A. (1985). Probability and fuzzy-set applications to information retrieval. *Annual review conformation science and technology*, 20, 117-151.

Bridger, R. S. (2003). *Introduction to ergonomics*. New York: Taylor and Francis.

Chatterjee, S. (1998). A connectionist approach for classifying accident narratives. Unpublished Ph.D. Dissertation. Purdue University, West Lafayette, IN.

Dewdney, A. K. (1997). Yes, We Have No Neutrons: An Eye-Opening Tour through the Twists and Turns of Bad Science.

Diaper, D. and Stanton, N. A. (2004). *The handbook of task analysis for human-computer interaction*. Mahwah, NJ: Lawrence Erlbaum.

Elm, W.C., Potter, S.S, and Roth E.M. (2003). Applied Cognitive Work Analysis: A Pragmatic Methodology for Designing Revolutionary Cognitive Affordances. In Hollnagel, E. (Ed.), *Handbook of Cognitive Task Design*, 357-382. Mahwah, NJ: Erlbaum.

Evans, G.W., Wilhelm, M.R., and Karwowski, W. (1987). A layout design heuristic employing the theory of fuzzy sets. *International Journal of Production Research*, 25, 1431-1450.

Fujihara, H., Simmons, D., Ellis, N., and Shannon, R. (1997). Knowledge conceptualization tool. *IEEE Transactions on Knowledge and Data Engineering*, 9, 209-220.

Gael, S. (1988). *The Job analysis handbook for business, industry, and government*, Vol. I and Vol. II. New York: John Wiley & Sons.

Gramopadhye, A. and Thaker, J. (1999). Task Analysis. In Karwowski, W. and Marras, W. S. (Eds.), *The occupational ergonomics handbook*, 17, 297-329.

Hatakeyama, N., Furuta, K., and Nakata, K. (2003). Model of Intention Inference Using Bayesian Network. In Stephanidis, C. and Jacko, J.A. (Eds.). Human-Centered Computing (v2). Human-computer interaction: *proceedings of HCI International 2003*, 390-394. Mahwah, NJ: Lawrence Erlbaum.

Hollnagel, E. (2003). Prolegomenon to Cognitive Task Design. In Hollnagel, E. (Ed.), *Handbook of Cognitive Task Design*, 3-15. Mahwah, NJ: Erlbaum.

Hollnagel, E. (2006). Task Analysis: Why, What, and How. In Salvendy, G. (Eds.), *Handbook of Human Factors and Ergonomics* (3rd Ed.), 14, 373-383. Hoboken, NJ: Wiley.

Hutton, R.J.B., Miller, T.E., and Thordsen, M.L. (2003). Decision-Centered Design: Leveraging Cognitive Task Analysis in Design. In Hollnagel, E. (Ed.), *Handbook of Cognitive Task Design*, 383-416. Mahwah, NJ: Erlbaum.

Huang, T. M., Kecman, V., and Kopriva, I. (2006). *Kernel Based Algorithms for Mining Huge Data Sets, Supervised, Semi-supervised, and Unsupervised Learning*. Springer-Verlag.

Jagielska, I., Matthews, C., and Whitford, T. (1999). Investigation into the application of neural networks, fuzzy logic, genetic algorithms, and rough sets to automated knowledge acquisition for classification problem. *Neurocomputing*, 24, 37-54.

Kirwan, B. and Ainsworth, L. K. (Eds.) (1992). A guide to Task Analysis. Taylor and Francis.

Klein, G. (1998). Sources of power: How people make decisions. *MIT Press*. Cambridge, MA.

Lehto, M.R., Boose, J., Sharit, J., and Salvendy, G. (1992). Knowledge Acquisition. In Salvendy, G. (Eds.), *Handbook of Industrial Engineering* (2nd Ed.), 58, 1495-1545. New York: John Wiley & Sons.

Lehto, M.R. and Buck, J.R. (2008, in print). *An Introduction to Human Factors and Ergonomics for Engineers*. Mahwah, NJ: Lawrence Erlbaum.

Lehto, M.R. and Sorock, G.S. (1996). Machine learning of motor vehicle accident categories from narrative data. *Methods of Information in Medicine*, 35 (4/5), 309-316.

Leman, S. and Lehto, M.R. (2003). Interactive decision support system to predict print quality. *Ergonomics*, 46(1-3), 52-67.

Lin, S. and Lehto, M.R. (2007). A Fuzzy Bayesian Model Based Semi-Automated Task Analysis, Human Interface, Part I, *HCII 2007*, 697–704. M.J. Smith, G. Salvendy (Eds.).

Lin, S. (2006). A Fuzzy Bayesian Model Based Semi-Automated Task Analysis. Unpublished Ph.D. Dissertation. Purdue University, West Lafayette, IN.

Liu, Y. (1997). Software-User Interface Design. In Salvendy, G. (Eds.), *Handbook of Human Factors and Ergonomics* (2nd Ed.), 51, 1689-1724. New York: John Wiley & Sons.

Luczak, H., Kabel, T, and Licht, T. (2006). Task Design and Motivation. In Salvendy, G. (Eds.), *Handbook of Human Factors and Ergonomics* (3rd Ed.), 15, 384-427. Hoboken, NJ: Wiley.

McCarthy, P. (2002). Machine Learning Applications for Pattern Recognition Within Call Center Data. Unpublished master thesis. Purdue University, West Lafayette, IN.

Morgeson, F.P., Medsker,G.J., and Campion M.A. (2006). Job and team design. In Salvendy, G. (Eds.), *Handbook of Human Factors and ergonomics* (3rd Ed.), 16, 428-457. Hoboken, NJ: Wiley.

Noorinaeini, A. and Lehto, M.R. (2007). Hybrid Singular Value Decomposition; a Model of Human Text Classification, Human Interface, Part I, *HCII 2007*, 517 – 525. M.J. Smith, G. Salvendy (Eds.).

Qiu, Shijun and Agogino, A. M. (2001). A Fusion of Bayesian and Fuzzy Analysis for Print Faults Diagnosis. *Proceedings of the International Society for Computers and Their Application-ISCA 16th International Conference*, 229-232.

Salvendy, G. (Eds.) (2006). *Handbook of Human Factors and ergonomics* (3rd Ed.). Hoboken, NJ: Wiley.

Schraagen, J.M., Chipman, S.F., and Shalin, V.L. (2000). *Cognitive task analysis*. Mahwah, NJ: Lawrence Erlbaum.

Shadbolt, N. and Burton, M. (2005). Knowledge Elicitation. In Wilson, J.R. and Corlett, E.N. (Eds.), *Evaluation of human work* (3rd Ed.), 14, 406-438. Taylor & Francis.

Shalin, V.L., Wisniewski, E.J., Levi, K.R., and Scott, P.D. (1988). A Formal Analysis of Machine Learning Systems for Knowledge Acquisition. *International Journal of Man-Machine Studies*, 29(4), 429-466.

Sheridan, T.B. (1997). Task analysis, task allocation and supervisory control. In Helander, M., Landauer, T.K. and Prabhu, P.V. (Eds.), *Handbook of human-computer interaction* (2nd Ed.), 87-105. Elsevier Science.

Stammers, R.B. and Shephard, A. (2005). Task Analysis. In Wilson, J.R. and Corlett, E.N. (Eds.), *Evaluation of human work* (3rd Ed.), 6, 144-168. Taylor & Francis.

Stephanidis, C. and Jacko, J.A. (Eds.) (2003). Human-Centred Computing (v2). Human-computer interaction: *proceedings of HCI International 2003*. Mahwah, NJ: Lawrence Erlbaum.

Todd, P. and Gigerenzer, G. (2001). Putting Naturalistic Decision Making into the Adaptive Toolbox, *Journal of Behavioral Decision Making*, 14, 353-384.

Trafton, J., Kirschenbaum, S., Tsui, T., Miyamoto, R., Ballas, J., Raymond, P. (2000). Turning pictures into numbers: extracting and generating information from complex visualizations, *International Journals of Human-Computer Studies*, 53, 827-850.

Vicente, K.J. (1999). *Cognitive Work Analysis: Toward Safe, Productive, and Healthy Computer-Based Work*. Mahwah, NJ: Lawrence Erlbaum.

Wellman, H.M., Lehto, M., Sorock, G.S., and Smith, G.S. (2004). Computerized Coding of Injury Narrative Data from the National Health Interview Survey, *Accident Analysis and Prevention*, 36(2), 165-171

Wickens, C.D. and Hollands J.G. (2000). *Engineering psychology and human performance*. New Jersey: Prentice Hall.

Wilson, J.R. and Corlett, E.N. (Eds.) (2005). *Evaluation of human work* (3rd Ed.). Taylor & Francis.

Wong, A.K.C. & Wang, Y. (2003). Pattern discovery: a data driven approach to decision support. *IEEE Transactions on Systems Man and Cybernetics, C*, 33(1), 114-124.

Yamamoto, H. and Sagisaka, Y. (1999). Multi-Class Composite N-gram based on connection direction. *Proceedings IEEE International Conference on Acoustics, Speech & Signal Processing*, 1, 533-536.

Zhu, W. and Lehto, M.R. (1999). Decision support for indexing and retrieval of information in hypertext system. *International Journal of Human Computer Interaction*, 11, 349-371.

Zsambok, C.E. (1997). Naturalistic Decision Making: Where Are We Now? In Zsambok, C.E. and Klein, G. A. (Eds.), *Naturalistic decision making*, 1, 3-16. Mahwah, NJ: Lawrence Erlbaum.

Zhou, H. and Huang, T.S. (2003). A Bayesian Framework for Real-Time 3D Hand Tracking in High Clutter Background. In Jacko, J.A. and Stephanidis, C. (Eds.). Human-Centered Computing (v1). Human-computer interaction: *proceedings of HCI International 2003*, 1303-1307. Mahwah, NJ: Lawrence Erlbaum.

KEY TERMS

Bayesian Inference: A type of statistical inference that uses Bayes' Rule to compute the posterior probability that the hypothesis is true, given the various sources of evidence is present.

Naïve Bayesian: A type of Bayesian inference that assumes evidences are conditionally independent given that the hypothesis is true and seeks to aggregate these evidences.

Database Indexing: A type of data structure that groups and classifies the data according to their criteria such as subject, words, and/or cross index for rapid access to records in the database.

Fuzzy Bayesian: A type of Bayesian inference that makes dependence assumptions of the evidences and reflects on the strongest evidence presented with no consideration given to negative evidence while co-occurrence of positive evidence is not aggregated.

Knowledge Acquisition: Means of extracting human thought processes as input data essential for the construction of knowledge-based systems

Machine Learning: Techniques and algorithms that allow computers to learn from recorded data rather than through direct acquisition of knowledge from human experts.

Naturalistic Decision Making: People use their experience to make decisions in field settings that often involve uncertain, dynamic, and information-rich problems with time constraints.

Task Analysis: A systematic identification and decomposition of a user's task into a number of small task components.

Text Mining: The process of parsing, filtering, categorizing, clustering, and analyzing the text to extract the relevance, usefulness, interestingness, and novelty of the text.

ENDNOTE

1 Figure 1 is cited and revised from Lin (2006).
2 Figure 2 is cited and revised from Lin (2006).

Behavioral Pattern–Based Customer Segmentation

Yinghui Yang
University of California, Davis, USA

INTRODUCTION

Customer segmentation is the process of dividing customers into distinct subsets (segments or clusters) that behave in the same way or have similar needs. Because each segment is fairly homogeneous in their behavior and needs, they are likely to respond similarly to a given marketing strategy. In the marketing literature, market segmentation approaches have often been used to divide customers into groups in order to implement different strategies. It has been long established that customers demonstrate heterogeneity in their product preferences and buying behaviors (Allenby & Rossi 1999) and that the model built on the market in aggregate is often less efficient than models built for individual segments. Much of this research focuses on examining how variables such as demographics, socioeconomic status, personality, and attitudes can be used to predict differences in consumption and brand loyalty. Distance-based clustering techniques, such as k-means, and parametric mixture models, such as Gaussian mixture models, are two main approaches used in segmentation. While both of these approaches have produced good results in various applications, they are not designed to segment customers based on their behavioral patterns.

There may exist natural behavioral patterns in different groups of customers or customer transactions (e.g. purchase transactions, Web browsing sessions, etc.). For example, a set of behavioral patterns that distinguish a group of wireless subscribers may be as follows: Their call duration during weekday mornings is short, and these calls are within the same geographical area. They call from outside the home area on weekdays and from the home area on weekends. They have several "data" calls on weekdays.

The above set of three behavioral patterns may be representative of a group of consultants who travel frequently and who exhibit a set of common behavioral patterns. This example suggests that there may be natural clusters in data, characterized by a set of typical behavioral patterns. In such cases, appropriate "behavioral pattern-based segmentation" approaches can constitute an intuitive method for grouping customer transactions.

BACKGROUND

The related work can be categorized into the following groups.

Market Segmentation

Since the concept emerged in the late 1950s, segmentation has been one of the most researched topics in the marketing literature. There have been two dimensions of segmentation research: segmentation bases and methods. A segmentation basis is defined as a set of variables or characteristics used to assign potential customers to homogenous groups. Research in segmentation bases focuses on identifying effective variables for segmentation, such as socioeconomic status, loyalty, and price elasticity (Frank et al 1972). Cluster analysis has historically been the most well-known method for market segmentation (Gordon 1980). Recently, much of market segmentation literature has focused on the technology of identifying segments from marketing data through the development and application of finite mixture models (see Böhning (1995) for a review). In general model-based clustering (Fraley & Raftery 1998; Fraley & Raftery 2002), the data is viewed as coming from a mixture of probability distributions, each representing a different cluster.

Pattern-Based Clustering

The definition of pattern-based clustering can vary. Some use this term to refer to clustering of patterns, e.g. pictures and signals. Others discover patterns from

the objects they are clustering and use the discovered patterns to help clustering the objects. In the second scenario, the definition of a pattern can vary as well. Wang et al (2002) considers two objects to be similar if they exhibit a coherent pattern on a subset of dimensions. The definition of a pattern is based on the correlation between attributes of objects to be clustered. Some other approaches use itemsets or association rules (Agrawal et al., 1995) as the representation of patterns. Han et al., (1997) addresses the problem of clustering-related customer transactions in a market basket database. Frequent itemsets used to generate association rules are used to construct a weighted hypergraph. Each frequent itemset is a hyperedge in the weighted hypergraph, and the weight of the hyperedge is computed as the average of the confidences for all possible association rules that can be generated from the itemset. Then, a hypergraph partitioning algorithm from Karypis et al., (1997) is used to partition the items such that the sum of the weights of hyperedges that are cut due to the partitioning is minimized. The result is a clustering of items (not transactions) that occur together in the transactions. Finally, the item clusters are used as the description of the cluster and a scoring metric is used to assign customer transactions to the best item cluster. Fung et al., (2003) used itemsets for document clustering. The intuition of their clustering criterion is that there are some frequent itemsets for each cluster (topic) in the document set, and different clusters share few frequent itemsets. A frequent itemset is a set of words that occur together in some minimum fraction of documents in a cluster. Therefore, a frequent itemset describes something common to many documents in a cluster. They use frequent itemsets to construct clusters and to organize clusters into a topic hierarchy. Yiu & Mamoulis (2003) uses projected clustering algorithms to find clusters in hidden subspaces. They realized the analogy between mining frequent itemsets and discovering the relevant subspace for a given cluster. They find projected clusters by mining frequent itemsets. Wimalasuriya et al., (2007) applies the technique of clustering based on frequent-itemsets in the domain of bio-informatics, especially to obtain clusters of genes based on Expressed Sequence Tags that make up the genes. Yuan et al., (2007) discovers frequent itemsets from image databases and feeds back discovered patterns to tune the similarity measure in clustering.

One common aspect among various pattern-based clustering methods is to define the similarity and/or the difference of objects/patterns. Then the similarity and difference are used in the clustering algorithms. The similarity and difference can be defined pairwise (between a pair of objects), or globally (e.g. within a cluster or between clusters). In the main focus section, we focus on the ones that are defined globally and discuss how these pattern-based clustering methods can be used for segmenting customers based on their behavioral patterns.

MAIN FOCUS OF THE CHAPTER

Segmenting Customers Based on Behavioral Patterns

The systematic approach to segment customers or customer transactions based on behavioral patterns is one that clusters customer transactions such that behavioral patterns generated from each cluster, while similar to each other within the cluster, are very different from the behavioral patterns generated from other clusters. Different domains may have different representations for what behavioral patterns are and for how to define similarity and difference between sets of behavioral patterns. In the wireless subscribers example described in the introduction, rules are an effective representation for behavioral patterns generated from the wireless call data; however, in a different domain, such as time series data on stock prices, representations for patterns may be based on "shapes" in the time series. It is easy to see that traditional distance-based clustering techniques and mixture models are not well suited to learning clusters for which the fundamental characterization is a set of patterns such as the ones above.

One reason that behavioral pattern-based clustering techniques can generate natural clusters from customer transactions is that such transactions often have natural categories that are not directly observable from the data. For example, Web transactions may be for work, for entertainment, shopping for self, shopping for gifts, transactions made while in a happy mood and so forth. But customers do not indicate the situation they are in before starting a transaction. However, the set of patterns corresponding to transactions in each category will be different. Transactions at work may be quicker and more focused, while transactions for entertainment may be long and across a broader set of sites. Hence, grouping transactions such that the patterns generated

from each cluster are very different from those generated from another cluster may be an effective method for learning the natural categorizations.

Behavioral Pattern Representation: Itemset

Behavioral patterns first need to be represented properly before they can be used for clustering. In many application domains, itemset is a reasonable representation for behavioral patterns. We illustrate how to use itemsets to represent behavioral patterns by using Web browsing data as an example. Assume we are analyzing Web data at a session level (continuous clicks are grouped together to form a session for the purpose of data analysis.). Features are first created to describe the session. The features can include those about time (e.g., average time spent per page), quantity (e.g., number of sites visited), and order of pages visited (e.g., first site) and therefore include both categorical and numeric types. A conjunction of atomic conditions on these attributes (an "itemset") is a good representation for common behavioral patterns in the Web data. For example, {starting_time = morning, average_time_page < 2 minutes, num_cate gories = 3, total_time < 10 minutes} is a behavioral pattern that may capture a user's specific "morning" pattern of Web usage that involves looking at multiple sites (e.g., work e-mail, news, finance) in a focused manner such that the total time spent is low. Another common pattern for this (same) user may be {starting_time = night, most_visted_category = games}, reflecting the user's typical behavior at the end of the day.

Behavioral patterns from other domains (e.g. shopping patterns in grocery stores) can be represented in a similar fashion. The attribute and value pair (starting_time = morning) can be treated as an item, and the combination of such items form an itemset (or a pattern). When we consider a cluster that contains objects with similar behavior patterns, we expect these objects in the cluster share many patterns (a list of itemsets).

Clustering Based on Frequent Itemsets

Clustering based on frequent-itemsets is recognized as a distinct technique and is often categorized under frequent-pattern based clustering methods (Han & Kamber 2006). Even though not a lot of existing research in this area addresses the problem of clustering based on behavioral patterns, the methods can potentially be modified for this purpose. Wang et al (1999) introduces a clustering criterion suggesting that there should be many large items within a cluster and little overlapping of such items across clusters. They then use this criterion to search for a good clustering solution. Wang et al (1999) also points out that, for transaction data, methods using pairwise similarity, such as k-means, have problems in forming a meaningful cluster. For transactions that come naturally in collection of items, it is more meaningful to use item/rule-based methods. Since we can represent behavioral patterns into a collection of items, we can potentially modify Wang et al (1999) so that there are many large items within a cluster and little overlapping of such items across clusters. Yang et al (2002) addresses a similar problem as that in Wang et al (1999), and does not use any pairwise distance function. They study the problem of categorical data clustering and propose a global criterion function that tries to increase the intra-cluster overlapping of transaction items by increasing the height-to-width ratio of the cluster histogram. The drawback of Wang et al (1999) and Yang et al (2002) for behavioral pattern based clustering is that they are not able to generate a set of large itemsets (a collection of behavioral patterns) within a cluster. Yang & Padmanabhan (2003, 2005) define a global goal and use this goal to guide the clustering process. Compared to Wang et al (1999) and Yang et al (2002), Yang & Padmanabhan (2003, 2005) take a new perspective of associating itemsets with behavior patterns and using that concept to guide the clustering process. Using this approach, distinguishing itemsets are identified to represent a cluster of transactions. As noted previously in this chapter behavioral patterns describing a cluster are represented by a set of itemsets (for example, a set of two itemsets {weekend, second site = eonline.com} and {weekday, second site = cnbc.com}. Yang & Padmanabhan (2003, 2005) allow the possibility to find a set of itemsets to describe a cluster instead of just a set of items, which is the focus of other item/itemsets-related work. In addition, the algorithms presented in Wang et al (1999) and Yang et al (2002) are very sensitive to the initial seeds that they pick, while the clustering results in Yang & Padmanabhan (2003, 2005) are stable. Wang et al (1999) and Yang et al (2002) did not use the concept of pattern difference and similarity.

The Framework for Behavioral Pattern-Based Clustering

Consider a collection of customer transactions to be clustered $\{ T_1, T_2, \ldots, T_n \}$. A clustering C is a partition $\{ C_1, C_2, \ldots, C_k \}$ of $\{ T_1, T_2, \ldots, T_n \}$ and each C_i is a cluster. The goal is to maximize the difference between clusters and the similarity of transactions within clusters. In words, we cluster to maximize a quantity M, where M is defined as follows:

$$M(C_1, C_2, ..., C_k) = Difference\ (C_1, C_2, ..., C_k) + \sum_{i=1}^{k} Similarity\ (C_i)$$

Here we only give specific definition for the difference between two clusters. This is sufficient, since hierarchical clustering techniques can be used to cluster the transactions repeatedly into two groups in such a way that the process results in clustering the transactions into an arbitrary number of clusters (which is generally desirable because the number of clusters does not have to be specified up front). The exact definition of difference and similarity will depend on the specific representation of behavioral patterns. Yang & Padmanabhan (2003, 2005) focus on clustering customers' Web transactions and uses itemsets as the representation of behavioral patterns. With the representation given, the difference and similarity between two clusters are defined as follows:

For each pattern P_a considered, we calculate the support of this pattern in cluster C_i and the support of the pattern in cluster C_j, then compute the relative difference between these two support values and aggregate these relative differences across all patterns. The support of a pattern in a cluster is the proportion of the transactions containing that pattern in the cluster. The intuition behind the definition of difference is that the support of the patterns in one cluster should be different from the support of the patterns in the other cluster if the underlying behavioral patterns are different. Here we use the relative difference between two support values instead of the absolute difference. Yang & Padmanabhan (2007) proves that under certain natural distributional assumptions the difference metric above is maximized when the correct clusters are discovered.

Here, the goal of the similarity measure is to capture how similar transactions are *within* each cluster. The heuristic is that, if transactions are more similar to each other, then they can be assumed to share more patterns. Hence, one approach is to use the number of strong patterns generated as a proxy for the similarity. If itemsets are used to represent patterns, then the number of frequent itemsets in a cluster can be used as a proxy for similarity.

The Clustering Algorithm

The ideal algorithm will be one that maximizes M (defined in previous section). However, for the objective function defined above, if there are n transactions and two clusters that we are interested in learning, the number of possible clustering schemes to examine is 2^n. Hence, a heuristic approach is called for. Yang & Padmanabhan (2003, 2005) provide two different clustering algorithms. The main heuristic used in the hierarchical algorithm presented in Yang & Padmanabhan (2005) is as follows. For each pattern, the data is divided into two parts such that all records containing that pattern are in one cluster and the remaining are in the other cluster. The division maximizing the global objective M is chosen. Further divisions are conducted following similar heuristic. The experiments in Yang & Padmanabhan (2003, 2005) indicate that the behavioral pattern-based customer segmentation approach is highly effective.

FUTURE TRENDS

Firms are increasingly realizing the importance of understanding and leveraging customer-level data, and critical business decision models are being built upon analyzing such data. Nowadays, massive amount of data is being collected for customers reflecting their behavioral patterns, so the practice of analyzing such data to identify behavioral patterns and using the patterns discovered to facilitate decision making is becoming more and more popular. Utilizing behavioral patterns for segmentation, classification, customer retention, targeted marketing, etc. is on the research agenda. For different application domains, the representations of behavioral patterns can be different. Different algorithms need to be designed for different pattern representations in different domains. Also, given the representation of the behavioral patterns, similarity and difference may also need to be defined differently. These all call for more research in this field.

CONCLUSION

As mentioned in the introduction, the existence of natural categories of customer behavior is intuitive, and these categories influence the transactions observed. Behavioral pattern-based clustering techniques, such as the one described in this chapter, can be effective in learning such natural categories and can enable firms to understand their customers better and build more accurate customer models. A notable strength of the behavioral pattern-based approach is the ability to explain the clusters and the differences between clusters.

REFERENCES

Agrawal, R., Mannila, H., Srikant, R., Toivonen, H. & Verkamo, A. I. (1995). Fast Discovery of Association Rules. In *Advances in Knowledge Discovery and Data Mining*, AAAI Press.

Allenby, G. M. & Rossi, P. E. (1999). Marketing Models of Consumer Heterogeneity. *Journal of Econometrics, 89*, 57-78.

Böhning, D. (1995). A Review of Reliable Maximum Likelihood Algorithms for Semiparametric Mixture Models. Journal *of Statistical Planning and Inference, 47*, 5-28.

Fraley, C. & Raftery, A. E. (1998). *How many clusters? Which clustering method? - Answers via Model-Based Cluster Analysis* (Tech. Rep. No. 329). Department of Statistics, University of Washington.

Fraley, C. & Raftery, A. E. (2002). Model-Based Clustering, Discriminant Analysis, and Density Estimation. *Journal of the American Statistical Association, 97*, 611-631.

Frank, R. E., Massy, W.F. & Wind, Y. (1972). *Market Segmentation*. Englewood Cliffs, New Jersey: Prentice Hall.

Fung, B. C. M., Wang, K. & Ester, M. (2003). Hierarchical Document Clustering using Frequent Itemsets. In *Proceedings of the Third SIAM International Conference on Data Mining*.

Gordon, A. D. (1980). *Classification*. London: Chapman and Hall.

J. Han and M. Kamber. Data Mining: Concepts and Techniques, pages 440-444. Morgan Kaufmann Publishers, second edition, 2006.

Han, E., Karypis, G., Kumar, V. & Mobasher, B. (1997). Clustering based on association rule hypergraphs. In *Proceedings of the SIGMOD'97 Workshop on Research Issues in Data Mining and Knowledge Discovery*.

Karypis, G., Aggarwal, R., Kumar, V. & Shekhar, S. (1997). Multilevel hypergraph partitioning: application in VLSI domain. In *Proceedings of the ACM/IEEE Design Automation Conference*.

Wang, H., Yang, J., Wang, W. & Yu, P.S. (2002). Clustering by Pattern Similarity in Large Data Sets. In *Proceedings of ACM SIGMOD Conference*.

Wang, K., Xu, C. & Liu, B. (1999). Clustering Transactions Using Large Items. In *Proceedings of the 8th Int. Conf. on Information and Knowledge Management*.

Wimalasuriya D., Ramachandran, S. & Dou D. (2007). Clustering Zebrafish Genes Based on Frequent-Itemsets and Frequency Levels. In *Proceedings of Pacific-Asia Conference on Knowledge Discovery and Data Mining*.

Yang, Y., Guan, X. & You, J. (2002). CLOPE: A Fast and Effective Clustering Algorithm for Transactional Data. In *Proceedings of SIGKDD*.

Yang, Y. & Padmanabhan, B. (2003). Segmenting Customer Transactions Using a Pattern-Based Clustering Approach. In *Proceedings of The Third IEEE International Conference on Data Mining*.

Yang, Y. & Padmanabhan, B. (2005). GHIC: A Hierarchical Pattern Based Clustering Algorithm for Grouping Web Transactions. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 7, No. 9, pp. 1300-1304.

Yang, Y. & Padmanabhan, B. (2007). Pattern-Based Clustering Approaches for Customer Segmentation. University of California, Davis, Working Paper.

Yiu, M. L. & Mamoulis, N. (2003). Frequent-Pattern based Iterative Projected Clustering. In *Proceedings of The Third IEEE International Conference on Data Mining*.

Yuan, J., Wu, Y. & Yang, M. (2007). From frequent itemsets to semantically meaningful visual patterns, In *Proceedings of the 13th ACM SIGKDD*.

KEY TERMS

Customer/Market Segmentation: The process of dividing customers/market into distinct subsets (segments) that behave in the same way or have similar needs.

Gaussian Mixture Models: A method used for clustering. It assumes that data comes from a distribution that is a combination of several Gaussian distributions.

Itemset: A set of items. It's often used in association rule mining. The occurrence frequency of an itemset (a set of items) is the number of transactions that contain the itemset. A frequent itemset is one that occurs often (with high frequency or support).

K-Means Clustering Algorithm: An algorithm to cluster objects based on attributes into k partitions. It assigns each object to the cluster whose center is nearest. It follows several iterations of assignments until convergence.

Model-Based Clustering: A type of clustering method. The data is viewed as coming from a mixture of probability distributions, each representing a different cluster.

Segmentation Basis: A segmentation basis is defined as a set of variables or characteristics used to assign potential customers to homogenous groups.

Web Browsing Sessions: A Web browsing session contains a list of consecutive clicks within a span of 30 minutes.

Best Practices in Data Warehousing

Les Pang
University of Maryland University College, USA

INTRODUCTION

Data warehousing has been a successful approach for supporting the important concept of knowledge management—one of the keys to organizational success at the enterprise level. Based on successful implementations of warehousing projects, a number of lessons learned and best practices were derived from these project experiences. The scope was limited to projects funded and implemented by federal agencies, military institutions and organizations directly supporting them.

Projects and organizations reviewed include the following:

- Census 2000 Cost and Progress System
- Defense Dental Standard System
- Defense Medical Logistics Support System Data Warehouse Program
- Department of Agriculture Rural Development Data Warehouse
- Department of Defense (DoD) Computerized Executive Information System
- Department of Energy, Lawrence Livermore National Laboratory, Enterprise Reporting Workbench
- Department of Health and Human Services, Health Care Financing Administration (HFCA) Teraplex Integration Center
- Environmental Protection Agency (EPA) Envirofacts Warehouse
- Federal Bureau of Investigation (FBI) Investigative Data Warehouse
- Federal Credit Union
- Internal Revenue Service (IRS) Compliance Data Warehouse
- Securities and Exchange Commission (SEC) Data Warehouse
- U.S. Army Operational Testing and Evaluation Command
- U.S. Coast Guard Executive Information System
- U.S. Navy Type Commander's Readiness Management System

BACKGROUND

Data warehousing involves the consolidation of data from various transactional data sources in order to support the strategic needs of an organization. This approach links the various silos of data that is distributed throughout an organization. By applying this approach, an organization can gain significant competitive advantages through the new level of corporate knowledge.

Various agencies in the Federal Government attempted to implement a data warehousing strategy in order to achieve data interoperability. Many of these agencies have achieved significant success in improving internal decision processes as well as enhancing the delivery of products and services to the citizen. This chapter aims to identify the best practices that were implemented as part of the successful data warehousing projects within the federal sector.

MAIN THRUST

Each best practice (indicated in **boldface**) and its rationale are listed below. Following each practice is a description of illustrative project or projects (indicated in *italics*), which support the practice.

Ensure the Accuracy of the Source Data to Maintain the User's Trust of the Information in a Warehouse

The user of a data warehouse needs to be confident that the data in a data warehouse is timely, precise, and complete. Otherwise, a user that discovers suspect data in warehouse will likely cease using it, thereby reduc-

ing the return on investment involved in building the warehouse. Within government circles, the appearance of suspect data takes on a new perspective.

HUD Enterprise Data Warehouse - Gloria Parker, HUD Chief Information Officer, spearheaded data warehousing projects at the Department of Education and at HUD. The HUD warehouse effort was used to profile performance, detect fraud, profile customers, and do "what if" analysis. Business areas served include Federal Housing Administration loans, subsidized properties, and grants. She emphasizes that the *public trust* of the information is critical. Government agencies do not want to jeopardize our public trust by putting out bad data. Bad data will result in major ramifications not only from citizens but also from the government auditing arm, the General Accounting Office, and from Congress (Parker, 1999).

EPA Envirofacts Warehouse - The Envirofacts data warehouse comprises of information from 12 different environmental databases for facility information, including toxic chemical releases, water discharge permit compliance, hazardous waste handling processes, Superfund status, and air emission estimates. Each program office provides its own data and is responsible for maintaining this data. Initially, the Envirofacts warehouse architects noted some data integrity problems, namely, issues with accurate data, understandable data, properly linked data and standardized data. The architects had to work hard to address these key data issues so that the public can trust that the quality of data in the warehouse (Garvey, 2003).

U.S. Navy Type Commander Readiness Management System – The Navy uses a data warehouse to support the decisions of its commanding officers. Data at the lower unit levels is aggregated to the higher levels and then interfaced with other military systems for a joint military assessment of readiness as required by the Joint Chiefs of Staff. The Navy found that it was spending too much time to determine its readiness and some of its reports contained incorrect data. The Navy developed a user friendly, Web-based system that provides quick and accurate assessment of readiness data at all levels within the Navy. "The system collects, stores, reports and analyzes mission readiness data from air, sub and surface forces" for the Atlantic and Pacific Fleets. Although this effort was successful, the Navy learned that data originating from the lower levels still needs to be accurate. The reason is that a number of legacy systems, which serves as the source data for the warehouse, lacked validation functions (Microsoft, 2000).

Standardize the Organization's Data Definitions

A key attribute of a data warehouse is that it serves as "a single version of the truth." This is a significant improvement over the different and often conflicting versions of the truth that come from an environment of disparate silos of data. To achieve this singular version of the truth, there needs to be consistent definitions of data elements to afford the consolidation of common information across different data sources. These consistent data definitions are captured in a data warehouse's metadata repository.

DoD Computerized Executive Information System (CEIS) is a 4-terabyte data warehouse holds the medical records of the 8.5 million active members of the U.S. military health care system who are treated at 115 hospitals and 461 clinics around the world. The Defense Department wanted to convert its fixed-cost health care system to a managed-care model to lower costs and increase patient care for the active military, retirees and their dependents. Over 12,000 doctors, nurses and administrators use it. Frank Gillett, an analyst at Forrester Research, Inc., stated that, "What kills these huge data warehouse projects is that the human beings don't agree on the definition of data. Without that . . . all that $450 million [cost of the warehouse project] could be thrown out the window" (Hamblen, 1998).

Be Selective on What Data Elements to Include in the Warehouse

Users are unsure of what they want so they place an excessive number of data elements in the warehouse. This results in an immense, unwieldy warehouse in which query performance is impaired.

Federal Credit Union - The data warehouse architect for this organization suggests that users know which data they use most, although they will not always admit to what they use least (Deitch, 2000).

Select the Extraction-Transformation-Loading (ETL) Strategy Carefully

Having an effective ETL strategy that extracts data from the various transactional systems, transforms the data to a common format, and loads the data into a relational or multidimensional database is the key to a successful data warehouse project. If the ETL strategy is not effective, it will mean delays in refreshing the data warehouse, contaminating the data warehouse with dirty data, and increasing the costs in maintaining the warehouse.

IRS Compliance Warehouse supports research and decision support, allows the IRS to analyze, develop, and implement business strategies for increasing voluntary compliance, improving productivity and managing the organization. It also provides projections, forecasts, quantitative analysis, and modeling. Users are able to query this data for decision support.

A major hurdle was to transform the large and diverse legacy online transactional data sets for effective use in an analytical architecture. They needed a way to process custom hierarchical data files and convert to ASCII for local processing and mapping to relational databases. They ended up with developing a script program that will do all of this. ETL is a major challenge and may be a "showstopper" for a warehouse implementation (Kmonk, 1999).

Leverage the Data Warehouse to Provide Auditing Capability

An overlooked benefit of data warehouses is its capability of serving as an archive of historic knowledge that can be used as an audit trail for later investigations.

U.S. Army Operational Testing and Evaluation Command (OPTEC) is charged with developing test criteria and evaluating the performance of extremely complex weapons equipment in every conceivable environment and condition. Moreover, as national defense policy is undergoing a transformation, so do the weapon systems, and thus the testing requirements. The objective of their warehouse was to consolidate a myriad of test data sets to provide analysts and auditors with access to the specific information needed to make proper decisions.

OPTEC was having "fits" when audit agencies, such as the General Accounting Office (GAO), would show up to investigate a weapon system. For instance, if problems with a weapon show up five years after it is introduced into the field, people are going to want to know what tests were performed and the results of those tests. A warehouse with its metadata capability made data retrieval much more efficient (Microsoft, 2000).

Leverage the Web and Web Portals for Warehouse Data to Reach Dispersed Users

In many organizations, users are geographically distributed and the World Wide Web has been very effective as a gateway for these dispersed users to access the key resources of their organization, which include data warehouses and data marts.

U.S. Army OPTEC developed a Web-based front end for its warehouse so that information can be entered and accessed regardless of the hardware available to users. It supports the geographically dispersed nature of OPTEC's mission. Users performing tests in the field can be anywhere from Albany, New York to Fort Hood, Texas. That is why the browser client the Army developed is so important to the success of the warehouse (Microsoft, 2000).

DoD Defense Dental Standard System supports more than 10,000 users at 600 military installations worldwide. The solution consists of three main modules: Dental Charting, Dental Laboratory Management, and Workload and Dental Readiness Reporting. The charting module helps dentists graphically record patient information. The lab module automates the workflow between dentists and lab technicians. The reporting module allows users to see key information though Web-based online reports, which is a key to the success of the defense dental operations.

IRS Compliance Data Warehouse includes a Web-based query and reporting solution that provides high-value, easy-to-use data access and analysis capabilities, be quickly and easily installed and managed, and scale to support hundreds of thousands of users. With this portal, the IRS found that portals provide an effective way to access diverse data sources via a single screen (Kmonk, 1999).

Make Warehouse Data Available to All Knowledgeworkers (Not Only to Managers)

The early data warehouses were designed to support upper management decision-making. However, over time, organizations have realized the importance of knowledge sharing and collaboration and its relevance to the success of the organizational mission. As a result, upper management has become aware of the need to disseminate the functionality of the data warehouse throughout the organization.

IRS Compliance Data Warehouse supports a diversity of user types—economists, research analysts, and statisticians—all of whom are searching for ways to improve customer service, increase compliance with federal tax laws and increase productivity. It is not just for upper management decision making anymore (Kmonk, 1999).

Supply Data in a Format Readable by Spreadsheets

Although online analytical tools such as those supported by Cognos and Business Objects are useful for data analysis, the spreadsheet is still the basic tool used by most analysts.

U.S. Army OPTEC wanted users to transfer data and work with information on applications that they are familiar with. In OPTEC, they transfer the data into a format readable by spreadsheets so that analysts can really crunch the data. Specifically, pivot tables found in spreadsheets allows the analysts to manipulate the information to put meaning behind the data (Microsoft, 2000).

Restrict or Encrypt Classified/Sensitive Data

Depending on requirements, a data warehouse can contain confidential information that should not be revealed to unauthorized users. If privacy is breached, the organization may become legally liable for damages and suffer a negative reputation with the ensuing loss of customers' trust and confidence. Financial consequences can result.

DoD Computerized Executive Information System uses an online analytical processing tool from a popular vendor that could be used to restrict access to certain data, such as HIV test results, so that any confidential data would not be disclosed (Hamblen, 1998). Considerations must be made in the architecting of a data warehouse. One alternative is to use a roles-based architecture that allows access to sensitive data by only authorized users and the encryption of data in the event of data interception.

Perform Analysis During the Data Collection Process

Most data analyses involve completing data collection before analysis can begin. With data warehouses, a new approach can be undertaken.

Census 2000 Cost and Progress System was built to consolidate information from several computer systems. The data warehouse allowed users to perform analyses during the data collection process; something was previously not possible. The system allowed executives to take a more proactive management role. With this system, Census directors, regional offices, managers, and congressional oversight committees have the ability to track the 2000 census, which never been done before (SAS, 2000).

Leverage User Familiarity with Browsers to Reduce Training Requirements

The interface of a Web browser is very familiar to most employees. Navigating through a learning management system using a browser may be more user friendly than using the navigation system of a proprietary training software.

U.S. Department of Agriculture (USDA) Rural Development, Office of Community Development, administers funding programs for the Rural Empowerment Zone Initiative. There was a need to tap into legacy databases to provide accurate and timely rural funding information to top policy makers. Through Web accessibility using an intranet system, there were dramatic improvements in financial reporting accuracy and timely access to data. Prior to the intranet, questions such as "What were the Rural Development investments in 1997 for the Mississippi Delta region?" required weeks of laborious data gathering and analysis, yet yielded obsolete answers with only an 80 percent accuracy factor. Now, similar analysis takes only a few minutes to perform, and the accuracy of the data is as

high as 98 percent. More than 7,000 Rural Development employees nationwide can retrieve the information at their desktops, using a standard Web browser. Because employees are familiar with the browser, they did not need training to use the new data mining system (Ferris, 2003).

Use Information Visualization Techniques Such as Geographic Information Systems (GIS)

A GIS combines layers of data about a physical location to give users a better understanding of that location. GIS allows users to view, understand, question, interpret, and visualize data in ways simply not possible in paragraphs of text or in the rows and columns of a spreadsheet or table.

EPA Envirofacts Warehouse includes the capability of displaying its output via the EnviroMapper GIS system. It maps several types of environmental information, including drinking water, toxic and air releases, hazardous waste, water discharge permits, and Superfund sites at the national, state, and county levels (Garvey, 2003). Individuals familiar with Mapquest and other online mapping tools can easily navigate the system and quickly get the information they need.

Leverage a Data Warehouse to Support Disaster Recovery

A warehouse can serve as a centralized repository of key data that can be backed up and secured to ensure business continuity in the event of a disaster.

The *Securities and Exchange Commission* (SEC) tracks daily stock transactions for the entire country. To manage this transactional data, the agency established a disaster recovery architecture that is based on a data warehouse. (Sybase Corporation, 2007)

Provide a Single Access Point to Multiple Sources of Data as part of the War Against Terrorism

A data warehouse can be used in the war against terrorism by bring together a collection of diverse data thereby allowing authorities to "connect the dots," e.g., identify associations, trends and anomalies.

The FBI uses its *Investigative Data Warehouse* to access more than 47 sources of counterterrorism data including file information from the FBI and other agencies as well as public sources. The FBI's warehouse includes a feature called "alert capability" which automatically notifies users when a newly uploaded document meets their search criteria. (Federal Bureau of Investigation, 2007).

Involve the Users when Identifying Warehousing Requirements

Users were asked for their requirements for the health-care-based data warehousing by surveying business users at the Health Care Financing Administration. Specifically, sample queries were identified such as: Of the total number of hip fractures in a given year, how many patients had surgery? Of those who had surgery, how many had infections?

By being more responsive to user needs, this led to the successful launch of HCFA's *Teraplex Integration Center*. (Makulowich, 1999)

Use a Modular Architecture to Respond to Changes

The data warehouse is a dynamic structure that changes due to new requirements. A monolithic architecture often mean changes throughout the data warehousing structure. By using a modular architecture instead, one can isolate where changes are needed.

Lawrence Livermore National Laboratory of the Department of Energy uses a modular architecture for its *Enterprise Reporting Workbench Data Architecture* to maximize flexibility, control and self-sufficiency. (The Data Warehouse Institute, 2007).

FUTURE TRENDS

Data warehousing will continue to grow as long as there are disparate silos of data sources throughout an organization. However, the irony is that there will be a proliferation of data warehouses as well as data marts, which will not interoperate within an organization. Some experts predict the evolution toward a federated architecture for the data warehousing environment. For example, there will be a common staging area for

data integration and, from this source, data will flow among several data warehouses. This will ensure that the "single truth" requirement is maintained throughout the organization (Hackney, 2000).

Another important trend in warehousing is one away from historic nature of data in warehouses and toward real-time distribution of data so that information visibility will be instantaneous (Carter, 2004). This is a key factor for business decision-making in a constantly changing environment. Emerging technologies, namely service-oriented architectures and Web services, are expected to be the catalyst for this to occur.

CONCLUSION

An organization needs to understand how it can leverage data from a warehouse or mart to improve its level of service and the quality of its products and services. Also, the organization needs to recognize that its most valuable resource, the workforce, needs to be adequately trained in accessing and utilizing a data warehouse. The workforce should recognize the value of the knowledge that can be gained from data warehousing and how to apply it to achieve organizational success.

A data warehouse should be part of an enterprise architecture, which is a framework for visualizing the information technology assets of an enterprise and how these assets interrelate. It should reflect the vision and business processes of an organization. It should also include standards for the assets and interoperability requirements among these assets.

REFERENCES

AMS. (1999). *Military marches toward next-generation health care service: The Defense Dental Standard System*.

Carter, M. (2004). *The death of data warehousing*. Loosely Coupled.

Deitch, J. (2000). *Technicians are from Mars, users are from Venus: Myths and facts about data warehouse administratio*n (Presentation).

Federal Bureau of Investigation (2007). *Information Technology.*

Ferris, N. (1999). *9 hot trends for '99*. Government Executive.

Ferris, N. (2003). *Information is power*. Government Executive.

Gerber, C. (1996). Feds turn to OLAP as reporting tool. *Federal Computer Week*.

Hackney, D. (2000). Data warehouse delivery: The federated future. *DM Review*.

Garvey, P. (2003). *Envirofacts warehouse public access to environmental data over the Web* (Presentation).

Hamblen, M. (1998). Pentagon to deploy huge medical data warehouse. Computer World.

Kirwin, B. (2003). *Management update: Total cost of ownership analysis provides many benefits*. Gartner Research, IGG-08272003-01.

Kmonk, J. (1999). *Viador information portal provides Web data access and reporting for the IRS*. DM Review.

Makulowich, John (1999). *IBM, Microsoft Build Presence at Federal Data Warehousing Table.* Washington Technology.

Matthews, W. (2000). Digging digital gold. *Federal Computer Week*.

Microsoft Corporation. (2000). *OPTEC adopts data warehousing strategy to test critical weapons systems*.

Microsoft Corporation. (2000). *U.S. Navy ensures readiness using SQL Server*.

Parker, G. (1999). Data warehousing at the federal government: A CIO perspective. In *Proceedings from Data Warehouse Conference '99.*

PriceWaterhouseCoopers. (2001). *Technology forecast*.

SAS. (2000). *The U.S. Bureau of the Census counts on a better system*.

Schwartz, A. (2000). Making the Web Safe. *Federal Computer Week*.

Sybase Corporation (2007). *Customer Success Story: U.S. Securities and Exchange Commission.*

The Data Warehouse Institute (2007). *Best Practices Awards 2007.*

KEY TERMS

ASCII: American Standard Code for Information Interchange. Serves a code for representing English characters as numbers with each letter assigned a number from 0 to 127.

Data Warehousing: A compilation of data designed to for decision support by executives, managers, analysts and other key stakeholders in an organization. A data warehouse contains a consistent picture of business conditions at a single point in time.

Database: A collection of facts, figures, and objects that is structured so that it can easily be accessed, organized, managed, and updated.

Enterprise Architecture: A business and performance-based framework to support cross-agency collaboration, transformation, and organization-wide improvement.

Extraction-Transformation-Loading (ETL): A key transitional set of steps in migrating data from the source systems to the database housing the data warehouse. Extraction refers to drawing out the data from the source system, transformation concerns converting the data to the format of the warehouse and loading involves storing the data into the warehouse.

Geographic Information Systems: Map-based tools used to gather, transform, manipulate, analyze, and produce information related to the surface of the Earth.

Hierarchical Data Files: Database systems that are organized in the shape of a pyramid with each row of objects linked to objects directly beneath it. This approach has generally been superceded by relationship database systems.

Knowledge Management: A concept where an organization deliberately and comprehensively gathers, organizes, and analyzes its knowledge, then shares it internally and sometimes externally.

Legacy System: Typically, a database management system in which an organization has invested considerable time and money and resides on a mainframe or minicomputer.

Outsourcing: Acquiring services or products from an outside supplier or manufacturer in order to cut costs and/or procure outside expertise.

Performance Metrics: Key measurements of system attributes that is used to determine the success of the process.

Pivot Tables: An interactive table found in most spreadsheet programs that quickly combines and compares typically large amounts of data. One can rotate its rows and columns to see different arrangements of the source data, and also display the details for areas of interest.

Terabyte: A unit of memory or data storage capacity equal to roughly 1,000 gigabytes.

Total Cost of Ownership: Developed by Gartner Group, an accounting method used by organizations seeking to identify their both direct and indirect systems costs.

Bibliomining for Library Decision-Making

Scott Nicholson
Syracuse University School of Information Studies, USA

Jeffrey Stanton
Syracuse University School of Information Studies, USA

INTRODUCTION

Most people think of a library as the little brick building in the heart of their community or the big brick building in the center of a college campus. However, these notions greatly oversimplify the world of libraries. Most large commercial organizations have dedicated in-house library operations, as do schools; nongovernmental organizations; and local, state, and federal governments. With the increasing use of the World Wide Web, digital libraries have burgeoned, serving a huge variety of different user audiences. With this expanded view of libraries, two key insights arise. First, libraries are typically embedded within larger institutions. Corporate libraries serve their corporations, academic libraries serve their universities, and public libraries serve taxpaying communities who elect overseeing representatives. Second, libraries play a pivotal role within their institutions as repositories and providers of information resources. In the provider role, libraries represent in microcosm the intellectual and learning activities of the people who comprise the institution. This fact provides the basis for the strategic importance of library data mining: By ascertaining what users are seeking, bibliomining can reveal insights that have meaning in the context of the library's host institution.

Use of data mining to examine library data might be aptly termed *bibliomining.* With widespread adoption of computerized catalogs and search facilities over the past quarter century, library and information scientists have often used bibliometric methods (e.g., the discovery of patterns in authorship and citation within a field) to explore patterns in bibliographic information. During the same period, various researchers have developed and tested *data-mining techniques,* which are advanced statistical and visualization methods to locate nontrivial patterns in large datasets. Bibliomining refers to the use of these bibliometric and data-mining techniques to explore the enormous quantities of data generated by the typical automated library.

BACKGROUND

Forward-thinking authors in the field of library science began to explore sophisticated uses of library data some years before the concept of data mining became popularized. Nutter (1987) explored library data sources to support decision making but lamented that "the ability to collect, organize, and manipulate data far outstrips the ability to interpret and to apply them" (p. 143). Johnston and Weckert (1990) developed a data-driven expert system to help select library materials, and Vizine-Goetz, Weibel, and Oskins (1990) developed a system for automated cataloging based on book titles (see also Morris, 1992, and Aluri & Riggs, 1990). A special section of *Library Administration and Management*, "Mining your automated system," included articles on extracting data to support system management decisions (Mancini, 1996), extracting frequencies to assist in collection decision making (Atkins, 1996), and examining transaction logs to support collection management (Peters, 1996).

More recently, Banerjeree (1998) focused on describing how data mining works and how to use it to provide better access to the collection. Guenther (2000) discussed data sources and bibliomining applications but focused on the problems with heterogeneous data formats. Doszkocs (2000) discussed the potential for applying neural networks to library data to uncover possible associations between documents, indexing terms, classification codes, and queries. Liddy (2000) combined natural language processing with text mining to discover information in digital library collections. Lawrence, Giles, and Bollacker (1999) created a system to retrieve and index citations from works in digital libraries. Gutwin, Paynter, Witten, Nevill-Manning, and Frank (1999) used text mining to support resource discovery.

These projects all shared a common focus on improving and automating two of the core functions of a library: acquisitions and collection management. A few

authors have recently begun to address the need to support management by focusing on understanding library users: Schulman (1998) discussed using data mining to examine changing trends in library user behavior; Sallis, Hill, Jancee, Lovette, and Masi (1999) created a neural network that clusters digital library users; and Chau (2000) discussed the application of Web mining to personalize services in electronic reference.

The December 2003 issue of *Information Technology and Libraries* was a special issue dedicated to the bibliomining process. Nicholson presented an overview of the process, including the importance of creating a data warehouse that protects the privacy of users. Zucca discussed the implementation of a data warehouse in an academic library. Wormell; Suárez-Balseiro, Iribarren-Maestro, & Casado; and Geyer-Schultz, Neumann, & Thede used bibliomining in different ways to understand the use of academic library sources and to create appropriate library services.

We extend these efforts by taking a more global view of the data generated in libraries and the variety of decisions that those data can inform. Thus, the focus of this work is on describing ways in which library and information managers can use data mining to understand patterns of behavior among library users and staff and patterns of information resource use throughout the institution.

MAIN THRUST

Integrated Library Systems and Data Warehouses

Most managers who wish to explore bibliomining will need to work with the technical staff of their Integrated Library System (ILS) vendors to gain access to the databases that underlie the system and create a data warehouse. The cleaning, preprocessing, and anonymizing of the data can absorb a significant amount of time and effort. Only by combining and linking different data sources, however, can managers uncover the hidden patterns that can help them understand library operations and users.

Exploration of Data Sources

Available library data sources are divided into three groups for this discussion: data from the *creation of the library,* data from the *use of the collection,* and data *from external sources* not normally included in the ILS.

ILS Data Sources from the Creation of the Library System

Bibliographic Information

One source of data is the collection of bibliographic records and searching interfaces that represents materials in the library, commonly known as the Online Public Access Catalog (OPAC). In a digital library environment, the same type of information collected in a bibliographic library record can be collected as metadata. The concepts parallel those in a traditional library: Take an agreed-upon standard for describing an object, apply it to every object, and make the resulting data searchable. Therefore, digital libraries use conceptually similar bibliographic data sources to traditional libraries.

Acquisitions Information

Another source of data for bibliomining comes from acquisitions, where items are ordered from suppliers and tracked until they are received and processed. Because digital libraries do not order physical goods, somewhat different acquisition methods and vendor relationships exist. Nonetheless, in both traditional and digital library environments, acquisition data have untapped potential for understanding, controlling, and forecasting information resource costs.

ILS Data Sources from Usage of theLibrary System

User Information

In order to verify the identity of users who wish to use library services, libraries maintain user databases. In libraries associated with institutions, the user database is closely aligned with the organizational database. Sophisticated public libraries link user records through zip codes with demographic information in order to learn more about their user population. Digital libraries may or may not have any information about their users, based upon the login procedure required. No matter

what data are captured about the patron, it is important to ensure that the identification information about the patron is separated from the demographic information before this information is stored in a data warehouse; doing so protects the privacy of the individual.

Circulation and Usage Information

The richest sources of information about library user behavior are circulation and usage records. Legal and ethical issues limit the use of circulation data, however. A data warehouse can be useful in this situation, because basic demographic information and details about the circulation could be recorded without infringing upon the privacy of the individual.

Digital library services have a greater difficulty in defining circulation, as viewing a page does not carry the same meaning as checking a book out of the library, although requests to print or save a full text information resource might be similar in meaning. Some electronic full-text services already implement the server-side capture of such requests from their user interfaces.

Searching and Navigation Information

The OPAC serves as the primary means of searching for works owned by the library. Additionally, because most OPACs use a Web browser interface, users may also access bibliographic databases, the World Wide Web, and other online resources during the same session; all this information can be useful in library decision making. Digital libraries typically capture logs from users who are searching their databases and can track, through clickstream analysis, the elements of Web-based services visited by users. In addition, the combination of a login procedure and cookies allows the connection of user demographics to the services and searches they used in a session.

External Data Sources

Reference Desk Interactions

In the typical face-to-face or telephone interaction with a library user, the reference librarian records very little information about the interaction. Digital reference transactions, however, occur through an electronic format, and the transaction text can be captured for later analysis, which provides a much richer record than is available in traditional reference work. The utility of these data can be increased if identifying information about the user can be captured as well, but again, anonymization of these transactions is a significant challenge.

Item Use Information

Fussler and Simon (as cited in Nutter, 1987) estimated that 75 to 80% of the use of materials in academic libraries is in house. Some types of materials never circulate, and therefore, tracking in-house use is also vital in discovering patterns of use. This task becomes much easier in a digital library, as Web logs can be analyzed to discover what sources the users examined.

Interlibrary Loan and Other Outsourcing Services

Many libraries use interlibrary loan and/or other outsourcing methods to get items on a need-by-need basis for users. The data produced by this class of transactions will vary by service but can provide a window to areas of need in a library collection.

Applications of Bibliomining Through a Data Warehouse

Bibliomining can provide an understanding of the individual sources listed previously in this article; however, much more information can be discovered when sources are combined through common fields in a data warehouse.

Bibliomining to Improve Library Services

Most libraries exist to serve the information needs of users, and therefore, understanding the needs of individuals or groups is crucial to a library's success. For many decades, librarians have suggested works; market basket analysis can provide the same function through usage data in order to aid users in locating useful works. Bibliomining can also be used to determine areas of deficiency and to predict future user needs. Common areas of item requests and unsuccessful searches may point to areas of collection weakness. By looking for patterns in high-use items, librarians can better predict the demand for new items.

Virtual reference desk services can build a database of questions and expert-created answers, which can be used in a number of ways. Data mining could be used to discover patterns for tools that will automatically assign questions to experts based upon past assignments. In addition, by mining the question/answer pairs for patterns, an expert system could be created that can provide users an immediate answer and a pointer to an expert for more information.

Bibliomining for Organizational Decision Making Within the Library

Just as the user behavior is captured within the ILS, the behavior of library staff can also be discovered by connecting various databases to supplement existing performance review methods. Although monitoring staff through their performance may be an uncomfortable concept, tighter budgets and demands for justification require thoughtful and careful performance tracking. In addition, research has shown that incorporating clear, objective measures into performance evaluations can actually improve the fairness and effectiveness of those evaluations (Stanton, 2000).

Low-use statistics for a work may indicate a problem in the selection or cataloging process. Looking at the associations between assigned subject headings, call numbers, and keywords, along with the responsible party for the catalog record, may lead to a discovery of system inefficiencies. Vendor selection and price can be examined in a similar fashion to discover if a staff member consistently uses a more expensive vendor when cheaper alternatives are available. Most libraries acquire works both by individual orders and through automated ordering plans that are configured to fit the size and type of that library. Although these automated plans do simplify the selection process, if some or many of the works they recommend go unused, then the plan might not be cost effective. Therefore, merging the acquisitions and circulation databases and seeking patterns that predict low use can aid in appropriate selection of vendors and plans.

Bibliomining for External Reporting and Justification

The library may often be able to offer insights to their parent organization or community about their user base through patterns detected with bibliomining. In addition, library managers are often called upon to justify the funding for their library when budgets are tight. Likewise, managers must sometimes defend their policies, particularly when faced with user complaints. Bibliomining can provide the data-based justification to back up the anecdotal evidence usually used for such arguments.

Bibliomining of circulation data can provide a number of insights about the groups who use the library. By clustering the users by materials circulated and tying demographic information into each cluster, the library can develop conceptual user groups that provide a model of the important constituencies of the institution's user base; this grouping, in turn, can fulfill some common organizational needs for understanding where common interests and expertise reside in the user community. This capability may be particularly valuable within large organizations where research and development efforts are dispersed over multiple locations.

FUTURE TRENDS

Consortial Data Warehouses

One future path of bibliomining is to combine the data from multiple libraries through shared data warehouses. This merger will require standards if the libraries use different systems. One such standard is the COUNTER project (2004), which is a standard for reporting the use of digital library resources. Libraries working together to pool their data will be able to gain a competitive advantage over publishers and have the data needed to make better decisions. This type of data warehouse can power evidence-based librarianship, another growing area of research (Eldredge, 2000).

Combining these data sources will allow library science research to move from making statements about a particular library to making generalizations about librarianship. These generalizations can then be tested on other consortial data warehouses and in different settings and may be the inspiration for theories. Bibliomining and other forms of evidence-based librarianship can therefore encourage the expansion of the conceptual and theoretical frameworks supporting the science of librarianship.

Bibliomining, Web Mining, and Text Mining

Web mining is the exploration of patterns in the use of Web pages. Bibliomining uses Web mining as its base but adds some knowledge about the user. This aids in one of the shortcomings of Web mining — many times, nothing is known about the user. This lack still holds true in some digital library applications; however, when users access password-protected areas, the library has the ability to map some information about the patron onto the usage information. Therefore, bibliomining uses tools from Web usage mining but has more data available for pattern discovery.

Text mining is the exploration of the context of text in order to extract information and understand patterns. It helps to add information to the usage patterns discovered through bibliomining. To use terms from information science, bibliomining focuses on patterns in the data that label and point to the information container, while text mining focuses on the information within that container. In the future, organizations that fund digital libraries can look to text mining to greatly improve access to materials beyond the current cataloging/metadata solutions.

The quality and speed of text mining continues to improve. Liddy (2000) has researched the extraction of information from digital texts; implementing these technologies can allow a digital library to move from suggesting texts that might contain the answer to just providing the answer by extracting it from the appropriate text or texts. The use of such tools risks taking textual material out of context and also provides few hints about the quality of the material, but if these extractions were links directly into the texts, then context could emerge along with an answer. This situation could provide a substantial asset to organizations that maintain large bodies of technical texts, because it would promote rapid, universal access to previously scattered and/or uncataloged materials.

Example of Hybrid Approach

Hwang and Chuang (in press) have recently combined bibliomining, Web mining, and text mining in a recommender system for an academic library. They started by using data mining on Web usage data for articles in a digital library and combining that information with information about the users. They then built a system that looked at patterns between works based on their content by using text mining. By combing these two systems into a hybrid system, they found that the hybrid system provides more accurate recommendations for users than either system taken separately. This example is a perfect representation of the future of bibliomining and how it can be used to enhance the text-mining research projects already in progress.

CONCLUSION

Libraries have gathered data about their collections and users for years but have not always used those data for better decision making. By taking a more active approach based on applications of data mining, data visualization, and statistics, these information organizations can get a clearer picture of their information delivery and management needs. At the same time, libraries must continue to protect their users and employees from the misuse of personally identifiable data records. Information discovered through the application of bibliomining techniques gives the library the potential to save money, provide more appropriate programs, meet more of the users' information needs, become aware of the gaps and strengths of their collection, and serve as a more effective information source for its users. Bibliomining can provide the data-based justifications for the difficult decisions and funding requests library managers must make.

REFERENCES

Atkins, S. (1996). Mining automated systems for collection management. *Library Administration & Management, 10*(1), 16-19.

Banerjee, K. (1998). Is data mining right for your library? *Computer in Libraries, 18*(10), 28-31.

Chau, M. Y. (2000). Mediating off-site electronic reference services: Human-computer interactions between libraries and Web mining technology. *IEEE Fourth International Conference on Knowledge-Based Intelligent Engineering Systems & Allied Technologies, USA, 2* (pp. 695-699).

Chaudhry, A. S. (1993). Automation systems as tools of use studies and management information. *IFLA Journal, 19*(4), 397-409.

COUNTER (2004). *COUNTER: Counting online usage of networked electronic resources.* Retrieved from *http://www.projectcounter.org/about.html*

Doszkocs, T. E. (2000). *Neural networks in libraries: The potential of a new information technology.* Retrieved from *http://web.simmons.edu/~chen/nit/NIT%2791/027~dos.htm*

Eldredge, J. (2000). Evidence-based librarianship: An overview. *Bulletin of the Medical Library Association, 88*(4), 289-302.

Geyer-Schulz, A., Neumann, A., & Thede, A. (2003). An architecture for behavior-based library recommender systems. *Information Technology and Libraries, 22*(4), 165-174.

Guenther, K. (2000). Applying data mining principles to library data collection. *Computers in Libraries, 20*(4), 60-63.

Gutwin, C., Paynter, G., Witten, I., Nevill-Manning, C., & Frank, E. (1999). Improving browsing in digital libraries with keyphrase indexes. *Decision Support Systems, 2I,* 81-104.

Hwang, S., & Chuang, S. (in press). Combining article content and Web usage for literature recommendation in digital libraries. *Online Information Review.*

Johnston, M., & Weckert, J. (1990). Selection advisor: An expert system for collection development. *Information Technology and Libraries, 9*(3), 219-225.

Lawrence, S., Giles, C. L., & Bollacker, K. (1999). Digital libraries and autonomous citation indexing. *IEEE Computer, 32*(6), 67-71.

Liddy, L. (2000, November/December). Text mining. *Bulletin of the American Society for Information Science*, 13-14.

Mancini, D. D. (1996). Mining your automated system for systemwide decision making. *Library Administration & Management, 10*(1), 11-15.

Morris, A. (Ed.). (1992). *Application of expert systems in library and information centers*. London: Bowker-Saur.

Nicholson, S. (2003). The bibliomining process: Data warehousing and data mining for library decision-making. *Information Technology and Libraries, 22*(4), 146-151.

Nutter, S. K. (1987). Online systems and the management of collections: Use and implications. *Advances in Library Automation Networking, 1,* 125-149.

Peters, T. (1996). Using transaction log analysis for library management information. *Library Administration & Management, 10*(1), 20-25.

Sallis, P., Hill, L., Janee, G., Lovette, K., & Masi, C. (1999). A methodology for profiling users of large interactive systems incorporating neural network data mining techniques. *Proceedings of the 1999 Information Resources Management Association International Conference* (pp. 994-998).

Schulman, S. (1998). Data mining: Life after report generators. *Information Today, 15*(3), 52.

Stanton, J. M. (2000). Reactions to employee performance monitoring: Framework, review, and research directions. *Human Performance, 13,* 85-113.

Suárez-Balseiro, C. A., Iribarren-Maestro, I., Casado, E. S. (2003). A study of the use of the Carlos III University of Madrid Library's online database service in Scientific Endeavor. *Information Technology and Libraries, 22(*4), 179-182.

Vizine-Goetz, D., Weibel, S., & Oskins, M. (1990). Automating descriptive cataloging. In R. Aluri, & D. Riggs (Eds.), *Expert systems in libraries* (pp. 123-127). Norwood, NJ: Ablex Publishing Corporation.

Wormell, I. (2003). Matching subject portals with the research environment. *Information Technology and Libraries, 22*(4), 158-166.

Zucca, J. (2003). Traces in the clickstream: Early work on a management information repository at the University of Pennsylvania. *Information Technology and Libraries, 22*(4), 175-178.

KEY TERMS

Bibliometrics: The study of regularities in citations, authorship, subjects, and other extractable facets from scientific communication by using quantitative and visualization techniques. This study allows researchers to understand patterns in the creation and documented use of scholarly publishing.

Bibliomining: The application of statistical and pattern-recognition tools to large amounts of data associated with library systems in order to aid decision making or to justify services. The term *bibliomining* comes from the combination of bibliometrics and data mining, which are the two main toolsets used for analysis.

Data Warehousing: The gathering and cleaning of data from disparate sources into a single database, which is optimized for exploration and reporting. The data warehouse holds a cleaned version of the data from operational systems, and data mining requires the type of cleaned data that live in a data warehouse.

Evidence-Based Librarianship: The use of the best available evidence, combined with the experiences of working librarians and the knowledge of the local user base, to make the best decisions possible (Eldredge, 2000).

Integrated Library System: The automation system for libraries that combines modules for cataloging, acquisition, circulation, end-user searching, database access, and other library functions through a common set of interfaces and databases.

Online Public Access Catalog (OPAC): The module of the Integrated Library System designed for use by the public to allow discovery of the library's holdings through the searching of bibliographic surrogates. As libraries acquire more digital materials, they are linking those materials to the OPAC entries.

NOTE

This work is based on Nicholson, S., & Stanton, J. (2003). Gaining strategic advantage through bibliomining: Data mining for management decisions in corporate, special, digital, and traditional libraries. In H. Nemati & C. Barko (Eds.). *Organizational data mining: Leveraging enterprise data resources for optimal performance* (pp. 247–262). Hershey, PA: Idea Group.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 100-105, copyright 2005 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

Bioinformatics and Computational Biology

Gustavo Camps-Valls
Universitat de València, Spain

Alistair Morgan Chalk
Eskitis Institute for Cell and Molecular Therapies, Griffiths University, Australia

INTRODUCTION

Bioinformatics is a new, rapidly expanding field that uses computational approaches to answer biological questions (Baxevanis, 2005). These questions are answered by means of *analyzing* and *mining biological data*. The field of *bioinformatics* or *computational biology* is a multidisciplinary research and development environment, in which a variety of techniques from computer science, applied mathematics, linguistics, physics, and, statistics are used. The terms *bioinformatics* and *computational biology* are often used interchangeably (Baldi, 1998; Pevzner, 2000). This new area of research is driven by the wealth of data from high throughput genome projects, such as the *human genome sequencing project* (International Human Genome Sequencing Consortium, 2001; Venter, 2001). As of early 2006, 180 organisms have been sequenced, with the capacity to sequence constantly increasing. Three major DNA databases collaborate and mirror over 100 billion base pairs in Europe (EMBL), Japan (DDBJ) and the USA (Genbank.) The advent of high throughput methods for monitoring gene expression, such as microarrays (Schena, 1995) detecting the expression level of thousands of genes simultaneously. Such data can be utilized to establish gene function (*functional genomics*) (DeRisi, 1997). Recent advances in *mass spectrometry* and proteomics have made these fields high-throughput. Bioinformatics is an essential part of *drug discovery*, *pharmacology*, *biotechnology*, *genetic engineering* and a wide variety of other *biological research* areas.

In the context of these proceedings, we emphasize that *machine learning* approaches, such as neural networks, hidden Markov models, or kernel machines, have emerged as good mathematical methods for analyzing (i.e. classifying, ranking, predicting, estimating and finding regularities on) biological datasets (Baldi, 1998). The field of bioinformatics has presented challenging problems to the machine learning community and the algorithms developed have resulted in new biological hypotheses. In summary, with the huge amount of information a mutually beneficial knowledge feedback has developed between theoretical disciplines and the life sciences. As further reading, we recommend the excellent "*Bioinformatics: A Machine Learning Approac*h" (Baldi, 1998), which gives a thorough insight into topics, methods and common problems in Bioinformatics.

The next section introduces the most important subfields of bioinformatics and computational biology. We go on to discuss current issues in bioinformatics and what we see are future trends.

BACKGROUND

Bioinformatics is a wide field covering a broad range of research topics that can broadly be defined as the management and analysis of data from generated by biological research. In order to understand bioinformatics it is essential to be familiar with at least a basic understanding of biology. The *central dogma* of molecular biology: DNA (a string of As, Cs, Gs and Ts) encodes genes which are *transcribed* into RNA (comprising As, Cs, Gs and Us) which are then generally *translated* into proteins (a string of *amino acids* – also denoted by single letter codes). The physical structure of these amino acids determines the proteins structure, which determines its function. A range of textbooks containing exhaustive information is available from the NCBI's website (http://www.ncbi.nlm.nih.gov/).

Major topics within the field of *bioinformatics* and *computational biology* can be structured into a number of categories, among which: prediction of gene expression and protein interactions, genome assembly, sequence alignment, gene finding, protein structure prediction, and evolution modeling are the most active

for the data mining community. Each of these problems requires different tools, computational techniques and machine learning methods. In the following section we briefly describe the main objectives in these areas:

1. *Databases and ontologies*. The overwhelming array of data being produced by experimental projects is continually being added to a collection of databases. The primary databases typically hold raw data and submission is often a requirement for publication. Primary databases include: a) sequence databases such as *Genbank, EMBL* and *DDBJ*, which hold nucleic acid sequence data (DNA, RNA), b) microarray databases such as *ArrayExpress* (Parkinson *et. al.* 2005), c) literature databases containing links to published articles such as *PubMed* (http://www.pubmed.com), and d) *PDB* containing protein structure data. Derived databases, created by analyzing the contents of primary databases creating higher order information such as a) protein domains, families and functional sites (*InterPro*, http://www.ebi.ac.uk/interpro/, Mulder *et. al.* 2003), and b) gene catalogs providing data from many different sources (*GeneLynx*, http://www.genelynx.org, Lenhard *et. al.* 2001, *GeneCards*, http://www.genecards.org, Safran *et. al.* 2003). An essential addition is the *Gene Ontology* project (http://www.geneontology.org). The Gene Ontology Consortium (2000), which provides a controlled vocabulary to describe genes and gene product attributes.
2. *Sequence analysis*. The most fundamental aspect of bioinformatics is sequence analysis. This broad term can be thought of as the identification of biologically significant regions in DNA, RNA or protein sequences. Genomic sequence data is analyzed to identify genes that code for RNAs or proteins, as well as regulatory sequences involved in turning on and off of genes. Protein sequence data is analyzed to identify signaling and structural information such as the location of biological active site(s). A comparison of genes within or between different species can reveal *relationships* between the genes (i.e. functional constraints). However, manual analysis of DNA sequences is impossible given the huge amount of data present. Database searching tools such as *BLAST* (http://www.ncbi.nlm.nih.gov/BLAST, Altschul *et. al.* 1990) are used to *search* the databases for similar sequences, using knowledge about *protein evolution*. In the context of genomics, *genome annotation* is the process biological features in a sequence. A popular system is the *ensembl* system which produces and maintains automatic annotation on selected eukaryotic genomes (http://www.ensembl.org).
3. *Expression analysis*. The expression of genes can be determined by measuring mRNA levels with techniques including microarrays, expressed cDNA sequence tag (EST) sequencing, sequence tag reading (e.g., SAGE and CAGE), massively parallel signature sequencing (MPSS), or various applications of multiplexed in-situ hybridization. Recently the development of protein microarrays and high throughput *mass spectrometry* can provide a snapshot of the proteins present in a biological sample. All of these techniques, while powerful, are noise-prone and/or subject to bias in the biological measurement. Thus, a major research area in computational biology involves developing *statistical tools to separate signal from noise* in high-throughput gene expression (HT) studies. Expression studies are often used as a first step in the process of identifying genes involved in pathologies by comparing the expression levels of genes between different tissue types (e.g. breast cancer cells vs. normal cells.) It is then possible to apply *clustering* algorithms to the data to determine the properties of cancerous vs. normal cells, leading to classifiers to diagnose novel samples. For a review of the microarray approaches in cancer, see Wang (2005).
4. *Genetics and population analysis*. The genetic variation in the population holds the key to identifying disease associated genes. Common polymorphisms such as single nucleotide polymorphisms (SNPs), insertions and deletions (*indels*) have been identified and ~3 million records are in the HGVBase polymorphism database (Fredman et. al. 2004). The international HapMap project is a key resource for finding genes affecting health, disease, and drug response (The International HapMap Consortium, 2005).
5. *Structural bioinformatics*. A proteins amino acid sequence (*primary structure*), is determined from the sequence of the gene that encodes it. This structure uniquely determines its physical structure. Knowledge of structure is vital to

understand protein function. Techniques available to predict the structure depend on the level of similarity to previously determined protein structures. In *homology modeling*, for instance, structural information from a homologous protein is used as a template to predict the structure of a protein once the structure of a homologous protein is known. State of the art in protein prediction is regularly gauged at the CASP (Critical Assessment of Techniques for Protein Structure Prediction, http://predictioncenter.org/) meeting, where leading protein structure prediction groups predict the structure of proteins that are experimentally verified.

6. *Comparative genomics and phylogenetic analysis.* In this field, the main goal is to identify similarities and differences between sequences in different organisms, to trace the evolutionary processes that have occurred in the divergence of the genomes. In bacteria, the study of many different species can be used to identify virulence genes (Raskin *et. al*. 2006). *Sequence analysis* commonly relies on evolutionary information about a gene or gene family (e.g. sites within a gene that are highly conserved over time imply a functional importance of that site). The complexity of genome evolution poses many exciting challenges to developers of mathematical models and algorithms, who have developed novel algorithmic, *statistical* and mathematical techniques, ranging from exact, heuristics, fixed parameter and approximation algorithms. In particular, the use of probabilistic-based models, such as *Markov Chain Monte Carlo* algorithms and *Bayesian analysis,* has demonstrated good results.
7. *Text mining.* The number of published scientific articles is rapidly increasing, making it difficult to keep up without using automated approaches based on text-mining (Scherf *et. al*. 2005). The extraction of biological knowledge from unstructured free text is a highly complex task requiring knowledge from *linguistics*, *machine learning* and the use of experts in the subject field.
8. *Systems biology*. The integration of large amounts of data from complementary methods is a major issue in bioinformatics. Combining datasets from many different sources makes it possible to construct networks of genes and interactions between genes. Systems biology incorporates disciplines such as statistics, graph theory and network analysis. An applied approach to the methods of systems biology is presented in detail by Winzeler (2006).

Biological systems are complex and constantly in a state of flux, making the measurement of these systems difficult. Data used in bioinformatics today is inherently noisy and incomplete. The increasing integration of data from different experiments adds to this noise. Such datasets require the application of sophisticated machine learning tools with the ability to work with such data.

Machine Learning Tools

A complete suite of machine learning tools is freely available to the researcher. In particular, it is worth noting the extended use of classification and regression trees (Breiman, 1984), Bayesian learning, neural networks (Baldi, 1998), Profile Markov models (Durbin *et. al.* 1998), rule-based strategies (Witten, 2005), or kernel methods (Schölkopf, 2004) in bioinformatics applications. For the interested reader, the software *Weka* (http://www.cs.waikato.ac.nz/ml/weka/, Witten, 2005), provides an extensive collection of machine learning algorithms for data mining tasks in general, and in particular bioinformatics. Specifically, implementations of well-known algorithms such as trees, Bayes learners, supervised and unsupervised classifiers, statistical and advanced regression, splines and visualization tools.

Bioinformatics Resources

The explosion of bioinformatics in recent years has lead to a wealth of databases, prediction tools and utilities and *Nucleic Acids Research* dedicates an issue to databases and web servers annually. Online resources are clustered around the large genome centers such as the National Center for Biotechnology Information (NCBI, http://www.ncbi.nlm.nih.gov/) and European Molecular Biology Laboratory (EMBL, http://www.embl.org/), which provide a multitude of public databases and services. For a useful overview of available links see "The Bioinformatics Links Directory" (http://bioinformatics.ubc.ca/resources/links_directory/, Fox *et. al.* 2005) or http://www.cbs.dtu.dk/biolinks/.

In addition to the use of online tools for analysis, it is common practice for research groups to access these databases and utilities locally. Different communities of bioinformatics programmers have set up free and open source projects such as EMBOSS (http://emboss.sourceforge.net/, Rice et al., 2000), Bioconductor (http://www.bioconductor.org), BioPerl, (http://www.bioperl.org, Stajich *et al.*, 2002), BioPython (http://www.biopython.org), and BioJava (http://www.biojava.org), which develop and distribute shared programming tools and objects. Several integrated software tools are also available, such as Taverna for bioinformatics workflow and distributed systems (http://taverna.sourceforge.net/), or Quantum 3.1 for drug discovery (http://www.q-pharm.com/home/contents/drug_d/soft).

FUTURE TRENDS

We have identified a broad set of problems and tools used within bioinformatics. In this section we focus specifically on the future of machine learning within bioinformatics. While a large set of data mining or machine learning tools have captured attention in the field of bioinformatics, it is worth noting that in application fields where a *similarity* measure has to be built, either to classify, cluster, predict or estimate, the use of *support vector machines* (SVM) and *kernel methods* (KM) are increasingly popular. This has been especially significant in computational biology, due to performance in real-world applications, strong modularity, mathematical elegance and convenience. Applications are encountered in a wide range of problems, from the classification of tumors to the automatic annotation of proteins. Their ability to work with high dimensional data, to process and efficiently integrate non-vectorial data (strings and images), along with a natural and consistent mathematical framework, make them very suitable to solve various problems arising in computational biology.

Since the early papers using SVM in bioinformatics (Mukherjee *et. al.*, 1998; Jaakkola *et. al.*, 1999), the application of these methods has grown exponentially. More than a mere application of well-established methods to new datasets, the use of kernel methods in computational biology has been accompanied by new developments to match the specificities and the needs of bioinformatics, such as methods for *feature selection* in combination with the classification of high-dimensional data, the introduction of *string kernels* to process biological sequences, or the development of methods to *learn from several kernels simultaneously* ('composite kernels'). The reader can find references in the book by Schölkopf et al. (2004), and a comprehensive introduction in (Vert, 2006). Several kernels for structured data, such as *sequences*, *trees* or *graphs*, widely developed and used in computational biology, are also presented in detail by Shawe-Taylor and Cristianini (2004).

In the near future, developments in *transductive* learning (improving learning through exploiting unlabeled samples), refinements in *feature selection* and *string-based* algorithms, and design of *biological-based kernels* will take place. Certainly, many computational biology tasks are transductive and/or can be specified through optimizing *similarity* criteria on strings. In addition, exploiting the nature of these learning tasks through engineered and problem-dependent kernels may lead to improved performance in many biological domains. In conclusion, the development of new methods for specific problems under the paradigm of kernel methods is gaining popularity and how far these methods can be extended is an unanswered question.

CONCLUSION

In this chapter, we have given an overview of the field of bioinformatics and computational biology, with its needs and demands. We have exposed the main research topics and pointed out common tools and software to tackle existing problems. We noted that the paradigm of kernel methods can be useful to develop a consistent formalism for a number of these problems. The application of advanced machine learning techniques to solve problems based on biological data will surely bring greater insight into the functionality of the human body. What the future will bring is unknown, but the challenging questions answered so far and the set of new and powerful methods developed ensure exciting results in the near future.

REFERENCES

Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J. (1990) Basic local alignment search tool. J Mol Biol. Oct 5;215(3):403-10.

Baldi, P. and Brunak, S. *Bioinformatics: A Machine Learning Approac*h. MIT Press. (1998).

Baxevanis, A.D. and Ouellette, B.F.F., eds., *Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins*, third edition. Wiley, 2005.

DeRisi, J. L., Iyer, V. R., and Brown, P. O. (1997). Exploring the metabolic and genetic control of gene expression on a genomic scale. Science, 278(5338):680–686.

Durbin, R., Eddy, S., Krogh, A., and Mitchison, G. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids.* Cambridge University Press.

Fox, J.A., Butland, S.L., McMillan, S., Campbell, G. and Ouellette, B.F. (2005) Bioinformatics Links Directory: a compilation of molecular biology web servers. Nucleic Acids Res., 33:W3-24.

Fredman, D., Munns, G., Rios, D., Sjoholm, F., Siegfried, M., Lenhard, B., Lehvaslaiho, H., Brookes, A.J. HGVbase: a curated resource describing human DNA variation and phenotype relationships. Nucleic Acids Res. 2004 Jan 1;32:D516-9.

International Human Genome Sequencing Consortium (2001). Initial sequencing and analysis of the human genome. Nature, 409(6822):860–921.

Jaakkola, T. S., Diekhans, M., and Haussler, D. (1999). Using the Fisher kernel method to detect remote protein homologies. In Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology, pages 149–158. AAAI Press.

Lenhard, B., Hayes, W.S., Wasserman, W.W. (2001) GeneLynx: a gene-centric portal to the human genome. Genome Res. Dec;11(12):2151-7.

Mukherjee, S., Tamayo, P., Mesirov, J. P., Slonim, D., Verri, A., and Poggio, T. (1998). Support vector machine classification of microarray data. Technical Report 182, C.B.L.C. A.I. Memo 1677.

Mulder, N.J., Apweiler ., Attwood, T.K., Bairoch, A., Barrell, D., Bateman, A., Binns, D., Biswas, M., Bradley, P., Bork, P., Bucher, P., Copley, R.R., Courcelle, E., Das, U., Durbin, R., Falquet, L., Fleischmann, W., Griffiths-Jones, S., Haft, D., Harte, N., Hulo, N., Kahn, D., Kanapin, A., Krestyaninova, M., Lopez, R., Letunic, I., Lonsdale, D., Silventoinen, V., Orchard, S.E., Pagni, M., Peyruc, D., Ponting, C.P., Selengut, J.D., Servant, F., Sigrist, C.J.A., Vaughan, R. and Zdobnov E.M. (2003). The InterPro Database, 2003 brings increased coverage and new features. Nucl. Acids. Res. 31: 315-318.

Parkinson, H., Sarkans, U., Shojatalab, M., Abeygunawardena, N., Contrino, S., Coulson, R., Farne, A., Garcia Lara, G., Holloway, E., Kapushesky, M., Lilja, P., Mukherjee, G., Oezcimen, A., Rayner, T., Rocca-Serra, P., Sharma A., Sansone, S. and Brazma, A.. (2005) ArrayExpress-a public repository for microarray gene expression data at the EBI. Nucl. Acids Res., 33: D553-D555.

Pevzner, P. A. *Computational Molecular Biology: An Algorithmic Approach* The MIT Press, 2000.

Raskin, D.M., Seshadri, R., Pukatzki, S.U. and Mekalanos, J.J. (2006) Bacterial genomics and pathogen evolution. Cell. 2006 Feb 24;124(4):703-14.

Rice,P., Longden,I., and Bleasby, A. (2000) EMBOSS: The European Molecular Biology Open Software Suite. Trends in Genetics 16, (6) pp276—277.

Safran, M., Chalifa-Caspi, V., Shmueli, O., Olender, T., Lapidot, M., Rosen, N., Shmoish, M., Peter, Y., Glusman, G., Feldmesser, E., Adato, A., Peter, I., Khen, M., Atarot, T., Groner, Y., Lancet, D. (2003) Human Gene-Centric Databases at the Weizmann Institute of Science: GeneCards, UDB, CroW 21 and HORDE. Nucleic Acids Res. Jan 1;31(1):142-6.

Scherf, M., Epple, A. and Werner, T. (2005) The next generation of literature analysis: integration of genomic analysis into text mining. Brief Bioinform. 2005 Sep;6(3):287-97.

Schölkopf, B., Tsuda, K., and Vert, J.-P. (2004). Kernel Methods in Computational Biology. MIT Press.

Shawe-Taylor, J. and Cristianini, N. (2004). Kernel Methods for Pattern Analysis. Cambridge University Press.

Stajich, J.E., Block, D., Boulez, K., Brenner, S.E., Chervitz, S.A., Dagdigian, C., Fuellen, G., Gilbert, J.G., Korf, I., Lapp, H., Lehvaslaiho, H., Matsalla, C., Mungall, C.J., Osborne, B.I., Pocock, M.R., Schattner, P., Senger, M., Stein, L.D., Stupka, E., Wilkinson, M.D., and Birney, E. *The Bioperl toolkit: Perl modules for the life sciences.* Genome Res 2002 Oct; 12(10) 1611-8.

The Gene Ontology Consortium (2000) Gene Ontology: tool for the unification of biology. Nature Genet. 25: 25-29.

The International HapMap Consortium. (2005) A haplotype map of the human genome. Nature 437, 1299-1320. 2005.

Venter, J. C. e. a. (2001). The sequence of the human genome. Science, 291(5507):1304–1351.

Vert, Jean-Philippe (2006). *Kernel methods in genomics and computational biology.* In book: "Kernel methods in bioengineering, signal and image processing". Eds.: G. Camps-Valls, J. L Rojo-Álvarez, M. Martínez-Ramón. Idea Group, Inc. Hershey, PA. USA.

Wang Y. (2005) Curr Opin Mol Ther. Jun;7(3):246-50.

Winzeler, E. A. (2006) Applied systems biology and malaria. Nature. Feb; 4:145-151.

Witten, I. H. and Frank, E. (2005) *Data Mining: Practical machine learning tools and techniques*, 2nd Edition, Morgan Kaufmann, San Francisco, 2005.

KEY TERMS

Alignment: A sequence alignment is a way of arranging DNA, RNA, or protein sequences so that similar regions are aligned. This may indicate functional or evolutionary relationships between the genes.

Functional Genomics: The use of the vast quantity of data produced by genome sequencing projects to describe genome function. Functional genomics utilizes high-throughput techniques such as microarrays to describe the function and interactions of genes.

Gene Expression: The process by which a gene's DNA sequence is converted into the structures and functions of a cell. Non-protein coding genes (e.g. rRNA genes) are not translated into protein.

Gene Ontology: A controlled vocabulary of terms to describe the function, role in biological processes and the location in the cell of a gene.

Homology: Similarity in sequence that is based on descent from a common ancestor.

Phylogenetics: The study of evolutionary relatedness among various groups of organisms usually carried out using alignments of a gene that is contained in every organism.

Protein Structure Prediction: The aim of determining the three-dimensional structure of proteins from their amino acid sequences.

Proteomics: The large-scale study of proteins, particularly their structures and functions.

Sequence Analysis: Analyzing a DNA or peptide sequence by sequence alignment, sequence searches against databases, or other bioinformatics methods.

Sequencing: To determine the primary structure (sequence) of a DNA, RNA or protein.

Similarity: How related one nucleotide or protein sequence is to another. The extent of similarity between two sequences is based on the percent of sequence identity and/or conservation.

Systems Biology: Integrating different levels of information to understand how biological systems function. The relationships and interactions between various parts of a biological system are modeled so that eventually model of the whole system can be developed. Computer simulation is often used.

Tertiary Structure: The three-dimensional structure of a polypeptide chain (protein) that results from the way that the alpha helices and beta sheets are folded and arranged.

Transcription: The process by which the information contained in a section of DNA is transferred to a newly assembled piece of messenger RNA (mRNA).

ENDNOTE

[1] Two genes are homologous if they originated from a gene in a common ancestral organism. Homology does not imply that the genes have the same function.

Biological Image Analysis via Matrix Approximation

Jieping Ye
Arizona State University, USA

Ravi Janardan
University of Minnesota, USA

Sudhir Kumar
Arizona State University, USA

INTRODUCTION

Understanding the roles of genes and their interactions is one of the central challenges in genome research. One popular approach is based on the analysis of microarray gene expression data (Golub *et al*., 1999; White, *et al*., 1999; Oshlack *et al*., 2007). By their very nature, these data often do not capture spatial patterns of individual gene expressions, which is accomplished by direct visualization of the presence or absence of gene products (mRNA or protein) (e.g., Tomancak *et al.*, 2002; Christiansen *et al.*, 2006). For instance, the gene expression pattern images of a *Drosophila melanogaster* embryo capture the spatial and temporal distribution of gene expression patterns at a given developmental stage (Bownes, 1975; Tsai *et al.*, 1998; Myasnikova *et al*., 2002; Harmon *et al.*, 2007). The identification of genes showing spatial overlaps in their expression patterns is fundamentally important to formulating and testing gene interaction hypotheses (Kumar *et al*., 2002; Tomancak *et al.*, 2002; Gurunathan *et al.*, 2004; Peng & Myers, 2004; Pan *et al.*, 2006).

Recent high-throughput experiments of *Drosophila* have produced over fifty thousand images (http://www.fruitfly.org/cgi-bin/ex/insitu.pl). It is thus desirable to design efficient computational approaches that can automatically retrieve images with overlapping expression patterns. There are two primary ways of accomplishing this task. In one approach, gene expression patterns are described using a controlled vocabulary, and images containing overlapping patterns are found based on the similarity of textual annotations. In the second approach, the most similar expression patterns are identified by a direct comparison of image content, emulating the visual inspection carried out by biologists [(Kumar *et al*., 2002); see also www.flyexpress.net]. The direct comparison of image content is expected to be complementary to, and more powerful than, the controlled vocabulary approach, because it is unlikely that all attributes of an expression pattern can be completely captured via textual descriptions. Hence, to facilitate the efficient and widespread use of such datasets, there is a significant need for sophisticated, high-performance, informatics-based solutions for the analysis of large collections of biological images.

BACKGROUND

The identification of overlapping expression patterns is critically dependent on a pre-defined pattern similarity between the standardized images. Quantifying pattern similarity requires deriving a vector of features that describes the image content (gene expression and localization patterns). We have previously derived a binary feature vector (BFV) in which a threshold value of intensity is used to decide the presence or absence of expression at each pixel coordinate, because our primary focus is to find image pairs with the highest spatial similarities (Kumar *et al.*, 2002; Gurunathan *et al*., 2004). This feature vector approach performs quite well for detecting overlapping expression patterns from early stage images. However, the BFV representation does not utilize the gradations in the intensity of gene expression because it gives the same weight to all pixels with greater intensity than the cut-off value. As a result, small regions without expression or with faint expression may be ignored, and areas containing mere noise may influence image similarity estimates. Pattern similarity based on the vector of pixel intensities

(of expression) has been examined by Peng & Myers (2004), and their early experimental results appeared to be promising. Peng & Myers (2004) model each image using the Gaussian Mixture Model (GMM) (McLachlan& Peel, 2000), and they evaluate the similarity between images based on patterns captured by GMMs. However, this approach is computationally expensive.

In general, the number of features in the BFV representation is equal to the number of pixels in the image. This number is over 40,000 because the FlyExpress database currently scales all embryos to fit in a standardized size of 320×128 pixels (www.flyexpress.net). Analysis of such high-dimensional data typically takes the form of extracting correlations between data objects and discovering meaningful information and patterns in data. Analysis of data with continuous attributes (e.g., features based on pixel intensities) and with discrete attributes (e.g., binary feature vectors) pose different challenges.

Principal Component Analysis (PCA) is a popular approach for extracting low-dimensional patterns from high-dimensional, continuous-attribute data (Jolliffe, 1986; Pittelkow & Wilson, 2005). It has been successfully used in applications such as computer vision, image processing, and bioinformatics. However, PCA involves the expensive eigen-decomposition of matrices, which does not scale well to large databases. Furthermore, PCA works only on data in vector form, while the native form of an image is a matrix. We have recently developed an approach called "Generalized Low Rank Approximation of Matrices" (GLRAM) to overcome the limitations of PCA by working directly on data in matrix form; this has been shown to be effective for natural image data (Ye *et al.*, 2004; Ye, 2005).

Here, we propose expression similarity measures that are derived from the correlation information among all images in the database, which is an advancement over the previous efforts wherein image pairs were exclusively used for deriving such measures (Kumar *et al.*, 2002; Gurunathan *et al.*, 2004; Peng & Myers, 2004). In other words, in contrast to previous approaches, we attempt to derive data-dependent similarity measures in detecting expression pattern overlap. It is expected that data-dependent similarity measures will be more flexible in dealing with more complex expression patterns, such as those from the later developmental stages of embryogenesis.

MAIN FOCUS

We are given a collection of n gene expression pattern images $\{A_1, A_2, \cdots, A_n\} \in \Re^{r \times c}$, with r rows and c columns. GLRAM (Ye, 2005, Ye *et al.*, 2004) aims to extract low-dimensional patterns from the image dataset by applying two transformations $L \in \Re^{r \times u}$ and $R \in \Re^{c \times v}$ with orthonormal columns, that is, $L^T L = I_u$ and $R^T R = I_v$, where I_u and I_v are identity matrices of size u and v, respectively. Each image A_i is transformed to a low-dimensional matrix $M_i = L^T A_i R \in \Re^{u \times v}$, for $i = 1, ..., n$. Here, $u < r$ and $v < c$ are two pre-specified parameters.

In GLRAM, the optimal transformations L^* and R^* are determined by solving the following optimization problem:

$$(L^*, R^*) = \arg\max_{L,R:L^T L=I_u, R^T R=I_v} \sum_{i=1}^{n} \left\| L^T A_i R \right\|_F^2 .$$

Here, $\|\cdot\|_F$ denotes the Frobenius norm of a matrix (Golub & Van Loan, 1996). To the best of our knowledge, there is no closed-form solution to the above maximization problem. However, if one of the two matrices L and R is given, the other one can be readily computed. More specifically, if L is given, the optimal R is given by the top eigenvectors of the matrix

$$\sum_{i=1}^{n} A_i^T L L^T A_i ,$$

while for a given R, the optimal L is given by the top eigenvectors of the matrix

$$\sum_{i=1}^{n} A_i R R^T A_i^T .$$

This results in an iterative procedure for computing L and R in GLRAM. For the given L and R, the low-dimensional matrix is given by $M_i = L^T A_i R$.

The dissimilarity between two expression patterns A_i and A_j is defined to be $\left\| M_i - M_j \right\|_F = \left\| L^T (A_i - A_j) R \right\|_F$. That is, GLRAM extracts the similarity between images through the transformations L and R. A key difference between the similarity computation based on the M_i's and the direct similarity computation based on the A_i's lies in the pattern extraction step involved in GLRAM. The columns of L and R form the basis

for expression pattern images, while M_i keeps the coefficients for the i-th image. Let L_j and R_k denote the j-th and k-th columns of L and R, respectively. Then, $L_j \cdot R_j \in \Re^{r \times c}$, for $j = 1, ..., u$ and $k = 1, ..., v$, forms the basis images. Note that the principal components in Principal Component Analysis (PCA) form the basis images, also called eigenfaces (Turk & Pentland, 1991) in face recognition.

We have conducted preliminary investigations on the use of GLRAM for expression pattern images from early developmental stages, and we have found that it performs quite well. Before GLRAM is applied, the mean is subtracted from all images. Using $u = 20$ and $v = 20$ on a set of *301* images from stage range *7--8*, the relative reconstruction error defined as

$$\sum_{i=1}^{n} \left\| A_i - LM_i R^T \right\|_F^2 \Big/ \sum_{i=1}^{n} \left\| A_i \right\|_F^2$$

is about *5.34%*. That is, even with a compression ratio as high as $320 \times 128/(20 \times 20) \approx 100$, the majority of the information (*94.66%*) in the original data is preserved. This implies that the intrinsic dimensionality of these embryo images from stage range *7-8* is small, even though their original dimensionality is large (about *40000*). Applying PCA with a similar compression ratio, we get a relative reconstruction error of about *30%*. Thus, by keeping the 2D structure of images, GLRAM is more effective in compression than PCA. The computational complexity of GLRAM is linear in terms of both the sample size and the data dimensionality, which is much lower than that of PCA. Thus, GLRAM scales to large-scale data sets. Wavelet transform (Averbuch *et al.*, 1996) is a commonly used scheme for image compression. Similar to the GLRAM algorithm, wavelets can be applied to images in matrix representation. A subtle but important difference is that wavelets mainly aim to compress and reconstruct a single image with a small cost of basis representations, which is extremely important for image transmission in computer networks. Conversely, GLRAM aims to compress a set of images by making use of the correlation information between images, which is important for pattern extraction and similarity-based pattern comparison.

FUTURE TRENDS

We have applied GLRAM for gene expression pattern image retrieval. Our preliminary experimental results show that GLRAM is able to extract biologically meaningful features and is competitive with previous approaches based on BFV, PCA, and GMM. However, the entries in the factorized matrices in GLRAM are allowed to have arbitrary signs, and there may be complex cancellations between positive and negative numbers, resulting in weak interpretability of the model. Non-negative Matrix Factorization (NMF) (Lee & Seung, 1999) imposes the non-negativity constraint for each entry in the factorized matrices, and it extracts bases that correspond to intuitive notions of the parts of objects. A useful direction for further work is to develop non-negative GLRAM, which restricts the entries in the factorized matrices to be non-negative while keeping the matrix representation for the data as in GLRAM.

One common drawback of all methods discussed in this chapter is that, for a new query image, the pairwise similarities between the query image and all the images in the database need to be computed. This pairwise comparison is computationally prohibitive, especially for large image databases, and some ad hoc techniques, like pre-computing all pairwise similarities, are usually employed. Recall that in BFV (Kumar *et al.*, 2002; Gurunathan *et al.*, 2004), we derive a binary feature vector for each image, which indicates the presence or absence of expression at each pixel coordinate. This results in a binary data matrix where each row corresponds to a BFV representation of an image. Rank-one approximation of binary matrices has been previously applied for compression, clustering, and pattern extraction in high-dimensional binary data (Koyuturk *et al.*, 2005). It can also be applied to organize the data into a binary tree where all data are contained collectively in the leaves and each internal node represents a pattern that is shared by all data at this node (Koyuturk *et al.*, 2005). Another direction for future work is to apply binary matrix approximations to construct such a tree-structured representation for efficient image retrieval.

CONCLUSION

Identification of genes with overlapping patterns gives important clues about gene function and interaction. Recent high-throughput experiments have produced a large number of images. It is thus desirable to design computational approaches that can automatically retrieve images with overlapping expression patterns. The approach presented here (GLRAM) approximates a set of data matrices with matrices of low rank, thus avoiding the conversion of images into vectors. Experimental results on gene expression pattern images demonstrate its effectiveness in image compression and retrieval.

ACKNOWLEDGMENT

We thank Ms. Kristi Garboushian for editorial support. This research has been supported by grants from the National Institutes of Health and the National Science Foundation.

REFERENCES

Averbuch, A., Lazar, D., & Israeli, M. (1996). Image compression using wavelet transform and multiresolution decomposition. *IEEE Transactions on Image Processing, 5:1*, 4–15.

Bownes, M. (1975). A photographic study of development in the living embryo of *Drosophila melanogaster*. *Journal of Embryology and Experimental Morphology*, 33, 789–801.

Christiansen, J.H., Yang, Y., Venkataraman, S., Richardson, L., Stevenson, P., Burton, N., Baldock, R.A., & Davidson, D.R. (2006). EMAGE: a spatial database of gene expression patterns during mouse embryo development. *Nucleic Acids Research.* 34: D637.

Golub, G. H. & Van Loan, C. F. (1996). *Matrix Computations*. The Johns Hopkins University Press, Baltimore, MD, USA, 3rd edition.

Golub, T. *et al*. (1999). Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286 (5439), 531-537.

Gurunathan, R., Emden, B. V., Panchanathan, S., & Kumar, S. (2004). Identifying spatially similar gene expression patterns in early stage fruit fly embryo images: binary feature versus invariant moment digital representations. *BMC Bioinformatics*, 5(202).

Harmon, C., Ahammad, P., Hammonds, A., Weiszmann, R., Celniker, S., Sastry, S., & Rubin, G. (2007). Comparative analysis of spatial patterns of gene expression in *Drosophila melanogaster* imaginal discs. In *Proceedings of the Eleventh International Conference on Research in Computational Molecular Biology*, 533-547.

Jolliffe, I. T. (1986). *Principal Component Analysis*. Springer-Verlag, New York.

Kumar, S., Jayaraman, K., Panchanathan, S., Gurunathan, R., Marti-Subirana, A., & Newfeld, S. J. (2002). BEST: a novel computational approach for comparing gene expression patterns from early stages of *Drosophlia melanogaster* development. *Genetics*, 169, 2037–2047.

Koyuturk, M., Grama, A., and Ramakrishnan, M.-N. (2005). Compression, clustering, and pattern discovery in very high-dimensional discrete-attribute data sets. *IEEE Transactions on Knowledge and Data Engineering*, 17(4), 447–461.

Lee, D.D. & Seung, H.S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401, 788-791.

McLachlan, G., & Peel, D. (2000). *Finite Mixture Models.* Wiley.

Myasnikova, E., Samsonova, A., Samsonova, M., & Reinitz, J. (2002). Support vector regression applied to the determination of the developmental age of a *Drosophila* embryo from its segmentation gene expression patterns. *Bioinformatics*, 18, S87–S95.

Oshlack, A., Chabot, A.E., Smyth, G.K., & Gilad, Y. (2007). Using DNA microarrays to study gene expression in closely related species. *Bioinformatics*, 23:1235-1242.

Pan, J., Guilherme, A., Balan, R., Xing, E. P., Traina, A. J. M., & Faloutsos, C. (2006).

Automatic mining of fruit fly embryo images. In *Proceedings of the Twelfth ACM SIGKDD International*

Conference on Knowledge Discovery and Data Mining, 693–698.

Peng, H. & Myers, E. W. (2004). Comparing *in situ* mRNA expression patterns of *Drosophila* embryos. In *Proceedings of the Eighth International Conference on Research in Computational Molecular Biology*, 157–166.

Pittelkow, Y., & Wilson, S.R. (2005). Use of principal component analysis and the GE-biplot for the graphical exploration of gene expression data. *Biometrics*, 61(2):630-632.

Tomancak, P., Beaton, A., Weiszmann, R., Kwan, E., Shu, S., Lewis, S. E., Richards, S., Ashburner, M., Hartenstein, V., Celniker, S. E., & Rubin, G. M. (2002). Systematic determination of patterns of gene expression during *Drosophila* embryogenesis. *Genome Biology*, 3(12).

Tsai, C. C., Kramer, S. G., & Gergen, J. P. (1998). Pair-rule gene *runt* restricts *orthodenticle* expression to the presumptive head of the *Drosophila* embryo. *Developmental Genetics*, 23(1), 35–44.

Turk, M. & Pentland, A. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1), 71–86.

White, K.P., Rifkin, S.A., Hurban, P., & Hogness, D.S. (1999). Microarray Analysis of *Drosophila* Development During Metamorphosis. *Science*, 286(5447):2179-2184.

Ye, J. (2005). Generalized low rank approximations of matrices. *Machine Learning*, 61(1- 3):167-191.

Ye, J., Janardan, R., & Li, Q. (2004). GPCA: An efficient dimension reduction scheme for image compression and retrieval. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 354-363.

KEY TERMS

Compression Ratio: The ratio between the space needed to store the original data, and the space needed to store the compressed data.

Drosophila Melanogaster: A two-winged insect that belongs to the Order Diptera, the order of the flies. The species is commonly known as the fruit fly, and is one of the most widely used model organisms in biology, including studies in genetics, physiology and life history evolution.

Developmental Stage: A distinct phase in Embryogenesis, which is traditionally divided into a series of consecutive stages distinguished by morphological markers. In a high throughput experimental study, embryonic images have been grouped into six stage ranges, 1-3, 4-6, 7-8, 9-10, 11-12, and 13-16.

Dimensionality Reduction: The process of reducing the number of random variables under consideration, which can be divided into feature selection and feature extraction.

Embryogenesis: A process by which the embryo is formed and develops. It starts with the fertilization of the ovum, egg, which, after fertilization, is then called a zygote. The zygote undergoes rapid mitotic divisions, the formation of two exact genetic replicates of the original cell, with no significant growth (a process known as cleavage) and cellular differentiation, leading to development of an embryo.

Gene: A set of segments of nucleic acid that contains the information necessary to produce a functional RNA product in a controlled manner.

Gene Expression: A process by which a gene's DNA sequence is converted into functional proteins.

Bitmap Join Indexes vs. Data Partitioning

Ladjel Bellatreche
Poitiers University, France

INTRODUCTION

Scientific databases and data warehouses store large amounts of data ith several tables and attributes. For instance, the Sloan Digital Sky Survey (SDSS) astronomical database contains a large number of tables with hundreds of attributes, which can be queried in various combinations (Papadomanolakis & Ailamaki, 2004). These queries involve many tables using binary operations, such as joins. To speed up these queries, many optimization structures were proposed that can be divided into two main categories: *redundant structures* like materialized views, advanced indexing schemes (bitmap, bitmap join indexes, etc.) (Sanjay, Chaudhuri & Narasayya, 2000) and vertical partitioning (Sanjay, Narasayya & Yang 2004) and *non redundant structures* like horizontal partitioning (Sanjay, Narasayya & Yang 2004; Bellatreche, Boukhalfa & Mohania, 2007) and parallel processing (Datta, Moon, & Thomas, 2000; Stöhr, Märtens & Rahm, 2000). These optimization techniques are used either in a sequential manner ou combined. These combinations are done intra-structures: materialized views and indexes for redundant and partitioning and data parallel processing for no redundant. Materialized views and indexes *compete for the same resource representing storage*, and *incur maintenance overhead in the presence of updates* (Sanjay, Chaudhuri & Narasayya, 2000). None work addresses the problem of selecting combined optimization structures. In this paper, we propose two approaches; one for combining a non redundant structures horizontal partitioning and a redundant structure bitmap indexes in order to reduce the query processing and reduce the maintenance overhead, and another to exploit algorithms for vertical partitioning to generate bitmap join indexes. To facilitate the understanding of our approaches, for review these techniques in details.

Data partitioning is an important aspect of physical database design. In the context of relational data warehouses, it allows tables, indexes and materialised views to be partitioned into disjoint sets of rows and columns that are physically stored and accessed separately (Sanjay, Narasayya & Yang 2004). It has a significant impact on performance of queries and manageability of data warehouses. Two types of data partitioning are available: vertical and horizontal partitionings.

The vertical partitioning of a table T splits it into two or more tables, called, sub-tables or vertical fragment, each of which contains a subset of the columns in T. Since many queries access only a small subset of the columns in a table, vertical partitioning can reduce the amount of data that needs to be scanned to answer the query. Note that the key columns are *duplicated* in each vertical fragment, to allow "reconstruction" of an original row in T. Unlike horizontal partitioning, indexes or materialized views, in most of today's commercial database systems there is no native Database Definition Language (DDL) support for defining vertical partitions of a table (Sanjay, Narasayya & Yang 2004). The horizontal partitioning of an object (a table, a vertical fragment, a materialized view, and an index) is specified using a partitioning method (range, hash, list), which maps a given row in an object to a key partition. All rows of the object with the same partition number are stored in the same partition.

Bitmap index is probably the most important result obtained in the data warehouse physical optimization field (Golfarelli, Rizzi & Saltarelli, 2002). The bitmap index is more suitable for low cardinality attributes since its size strictly depends on the number of distinct values of the column on which it is built. Bitmap join indexes (BJIs) are proposed to speed up join operations (Golfarelli, Rizzi & Saltarelli, 2002). In its simplest form, it can be defined as a bitmap index on a table R based on a single column of another table S, where S commonly joins with R in a specific way.

Many studies have recommended the combination of redundant and non redundant structures to get a better performance for a given workload (Sanjay, Narasayya & Yang 2004; Bellatreche, Schneider, Lorinquer & Mohania, 2004). Most of previous work in physical database design did not consider the interdependence between redundant and no redundant optimization structures. Logically, BJIs and horizontal partitioning

are two similar optimization techniques - both speed up query execution, pre-compute join operations and concern selection attributes of dimension tables[1]. Furthermore, BJIs and HP can interact with one another, i.e., the presence of an index can make a partitioned schema more efficient and vice versa (since fragments have the same schema of the global table, they can be indexed using BJIs and BJIs can also be partitioned (Sanjay, Narasayya & Yang 2004)).

BACKGROUND

Note that each BJI can be defined on one or several non key dimension's attributes with a low cardinality (that we call indexed columns) by joining dimension tables owned these attributes and the fact table[2].

Definition: An indexed attribute A_j candidate for defining a BJI is a column A_j of a dimension table D_i with a low cardinality (like gender attribute) such that there is a selection predicate of the form: $D_i.A_j$ θ value, θ is one of six comparison operators {=,<,>,<=,>=}, and value is the predicate constant.

For a large number of indexed attributes candidates, selecting optimal BJIs is an NP-hard problem (Bellatreche, Boukhalfa & Mohania, 2007).

On the other hand, the best way to partition a relational data warehouse is to decompose the fact table based on the fragmentation schemas of dimension tables (Bellatreche & Boukhalfa, 2005). Concretely, (1) partition some/all dimension tables using their simple selection predicates ($D_i.A_j$ θ value), and then (2) partition the facts table using the fragmentation schemas of the fragmented dimension tables (this fragmentation is called derived horizontal fragmentation (Özsu a Valduriez, 1999)). This fragmentation procedure takes into consideration the star join queries requirements. The number of horizontal fragments (denoted by N) of the fact table generated by this partitioning procedure is given by:

$$N = \prod_{i=1}^{g} m_i,$$

where m_i and g are the number of fragments of the dimension table D_i and the number of dimension tables participating in the fragmentation process, respectively. This number may be very large (Bellatreche & Boukhalfa & Abdalla, 2006). Based on this definition, there is a strong similarity between BJIs and horizontal partitioning as show the next section.

Similarity between HP and BJIs

To show the similarity between HP and BJIs, the following scenario is considered[3]. Suppose a data warehouse represented by three dimension tables (TIME, CUSTOMER and PRODUCT) and one fact table (SALES). The population of this schema is given in Figure 1. On the top of this the following query is executed:

```
SELECT Count(*)
FROM CUSTOMER C, PRODUCT P, TIME T, SALES S
WHEERE C.City='LA'
AND P.Range='Beauty'
AND T.Month='June'
AND P.PID=S.PID
AND C.CID=S.CID
AND T.TID=S.TID
```

This query has three selection predicates defined on dimension table attributes City (City='LA'), Range (Range='Beauty') and Month (Month = 'June') and

Figure 1. Sample of data warehouse population

Customer

RID^C	CID	Name	City
6	616	Gilles	LA
5	515	Yves	Paris
4	414	Patrick	Tokyo
3	313	Didier	Tokyo
2	212	Eric	LA
1	111	Pascal	LA

Product

RID^P	PID	Name	Range
6	106	Sonoflore	Beauty
5	105	Clarins	Beauty
4	104	WebCam	Multimedia
3	103	Barbie	Toys
2	102	Manure	Gardering
1	101	SlimForm	Fitness

Time

RID^T	TID	Month	Year
6	11	Jan	2003
5	22	Feb	2003
4	33	Mar	2003
3	44	Apr	2003
2	55	May	2003
1	66	Jun	2003

Sales

RID^S	CID	PID	TID	Amount
1	616	106	11	25
2	616	106	66	28
3	616	104	33	50
4	545	104	11	10
5	414	105	66	14
6	212	106	55	14
7	111	101	44	20
8	111	101	33	27
9	212	101	11	100
10	313	102	11	200
11	414	102	11	102
12	414	102	55	103
13	515	102	66	100
14	515	103	55	17
15	212	103	44	45
16	111	105	66	44
17	212	104	66	40
18	515	104	22	20
19	616	104	22	20
20	616	104	55	20
21	212	105	11	10
22	212	105	44	10
23	212	105	55	18
24	212	106	11	18
25	313	105	66	19
26	313	105	22	17
27	313	106	11	15

three join operations. It can be executed using either horizontal partitioning (this strategy is called HPFIRST) or BJI (called BJIFIRST).

HPFIRST: The Database Administrator (DBA) partition the fact table using the derived horizontal fragmentation based on partitioning schemas of dimension tables: CUSTOMER, TIME and PRODUCT based on City (thee fragments), Month (six fragments), Range (five fragments), respectively (see Figure 2). Consequently, the fact table is fragmented in 90 fragments (3 * 6 * 5), where each fact fragment is defined as follows:

$Sales_i$ = SALES $\lceil$ $CUSTOMER_j$ $\lceil$ $TIME_k$ $\lceil$ $PRODUCT_m$

where $\lceil$ represents semi join operation, $(1 \le i \le 90)$, $(1 \le j \le 3)$, $(1 \le k \le 6)$, $(1 \le m \le 5)$.

Figure 2c shows the fact fragment SALES_BPJ corresponding to sales of beauty products realized by customers living at LA City during month of June.

To execute the aforementioned query, the optimizer shall rewrite it according to the fragments. The result of the rewriting process is:

```
SELECT Count(*) FROM SALES_BPJ
```

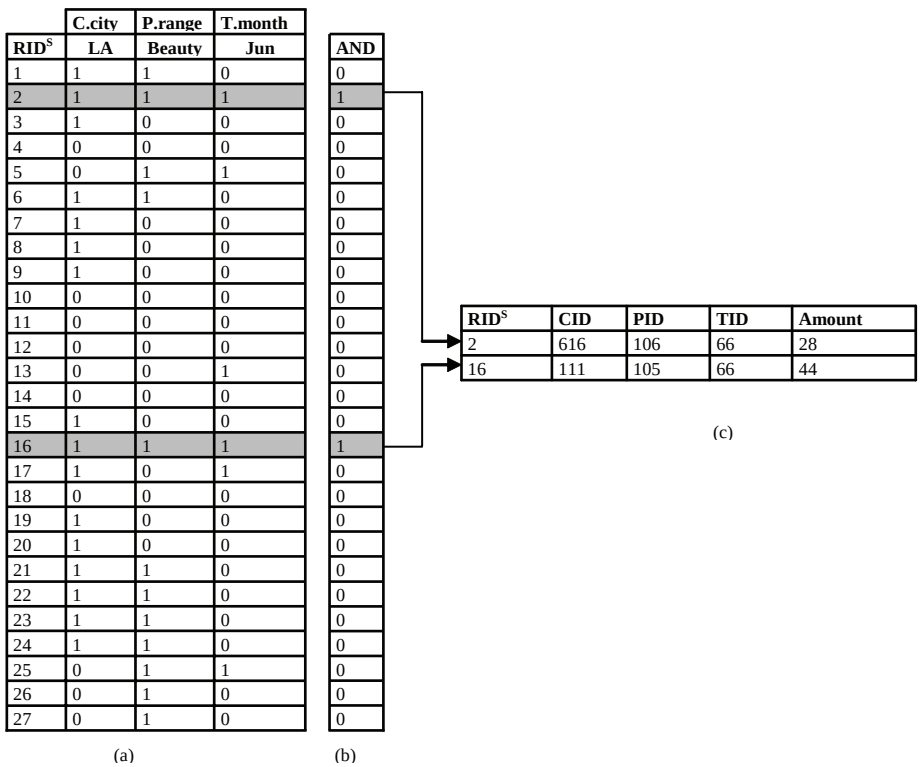
Instead of loading all tables (fact table and the three dimension tables) of the warehouse, the query optimizer loads only the fact fragment SALES_BPJ.

BJIFIRST: The DBA may define a BJI on three dimension attributes: City, Month and Range as follows:

```
CREATE BITMAP INDEX Sales_CPT_bjix
ON SALES(CUSTOMER.City, PRODUCT.Range,
TIME.Month)
FROM SALES S, CUSTOMER C, TIME T, PROD-
UCT P
WHERE S.CID= C.CID AND S.PID=P.PID AND
S.TID=T.TID
```

Figure 2a shows the population of the BJI Sales_CPT_bjix. Note that the three attributes: City, Month and Range are indexed attributes and fragmentation attributes. To execute the aforementioned query, the optimizer just accesses the bitmaps corresponding to the columns of Sales_CPT_bjx representing June, Beauty and LA and performs AND operation. This

Figure 2. (a) The bitmap join index, (b) the AND operation bits (c) The fact fragment



RID^S	C.city	P.range	T.month
	LA	Beauty	Jun
1	1	1	0
2	1	1	1
3	1	0	0
4	0	0	0
5	0	1	1
6	1	1	0
7	1	0	0
8	1	0	0
9	1	0	0
10	0	0	0
11	0	0	0
12	0	0	0
13	0	0	1
14	0	0	0
15	1	0	0
16	1	1	1
17	1	0	1
18	0	0	0
19	1	0	0
20	1	0	0
21	1	1	0
22	1	1	0
23	1	1	0
24	1	1	0
25	0	1	1
26	0	1	0
27	0	1	0

(a)

AND
0
1
0
0
0
0
0
0
0
0
0
0
0
0
0
0
1
0
0
0
0
0
0
0
0
0
0

(b)

RID^S	CID	PID	TID	Amount
2	616	106	66	28
16	111	105	66	44

(c)

example shows a strong similarity between horizontal partitioning and BJIs – both save three join operations and optimize selection operations defined on dimension tables.

Another similarity between BJI and horizontal partitioning resides on their selection problem formalisations.

The HP selection problem may be formulated as follows: (Bellatreche & Boukhalfa, 2005): Given a data warehouse with a set of dimension tables $D = \{D_1, D_2, ..., D_d\}$ and a fact table F, a workload Q of queries $Q = \{Q_1, Q_2, ..., Q_m\}$, where each query Q_i $(1 \leq i \leq m)$ has an access frequency f_i. The HP selection problem consists in fragmenting the fact table into a set of N fact fragments $\{F_1, F_2, ..., F_N\}$ such that: (1) the sum of the query processing cost when executed on top of the partitioned star schema is minimized and (2) $N \leq W$, where W is a threshold, fixed by the database administrator (DBA), representing the maximal number of fact fragments that she can maintain.

The BJI selection problem has the same formulation as HP, except the definition of its constraint (Bellatreche, Missaoui, Necir & Drias, 2007). The aim of BJI selection problem is to find a set of BJIs minimizing the query processing cost and storage requirements of BJIs do not exceed S (representing storage bound).

Differences

The main difference between the data partitioning and BJI concerns the storage cost and the maintenance overhead required for storing and maintaining (after INSERT, DELETE, or UPDATE operations) BJIs. To reduce the cost of storing BJIs, several works has been done on compression techniques (Johnson, 1999).

MAIN FOCUS

Horizontal Partitioning and BJIs

Actually, many commercial database systems support horizontal partitioning and BJIs. The use of these optimization structures is done in a sequential manner by DBA: (1) HPFIRST or (2) BJIFIRST. It will interesting to combine these two structures to optimize queries processing cost and to reduce the maintenance and storage cost. This combination, called HP→BJI is feasible since horizontal partitioning preserves the schema of base tables, therefore, the obtained fragments can further be indexed. Note that HP and BJIs are defined on selection attributes ($A = \{A1, \ldots, Ao\}$) of dimension tables defined on the queries. The main issue to combine these optimization techniques is to identify selection attributes (from A) for HP and for BJI. We propose a technique to combine them: Given a data warehouse schema with a fact table F and set of dimension tables $\{D_1, \ldots, D_l\}$. Let $Q = \{Q_1, Q_2, \cdots, Q_m\}$ be a set of most frequently queries.

1. Partition the database schema using any fragmentation algorithm (for example the genetic algorithm developed in (Bellatreche & Boukhalfa, 2005)) according a set of queries Q and the threshold W fixed by the DBA. Our genetic algorithm generates a set of N fragments of fact tables. Let FASET be the set of fragmentation attributes of dimension tables used in definition of fragments ($FASET \subseteq A$). In the motivating example (see Background), Month, City and Range represent fragmentation attributes. Among m queries, an identification of those that get benefit from HP, denoted by $Q' = \{Q'_1, Q'_2, \cdots, Q'_l\}$ is done. This identification thanks to a rate defined for each query Q_j of Q as follows:

$$rate(Q_j) = \frac{C[Q_j, FS]}{C[Q_j, \varphi]} \text{ where } C[Q_j, FS] \text{ and } C[Q_j, \phi]$$

represent the cost of executing the query Q_j on un-partitioned data warehouse and partitioned schema FS, respectively. The DBA has the right to set up this rate using a threshold λ: if rate $(Q_j) \leq \lambda$ then Q_j is a profitable query, otherwise no profitable query.

2. Among (A - FASET), pick up attribute(s) with a low cardinality in order to built BJIs. The set of these attributes is denoted by BJISET.
3. The selection of BJIs is then done using BJISET and no profitable queries $Q' = \{Q'_1, Q'_2, \cdots, Q'_l\}$ using a greedy algorithm (Bellatreche, Boukhalfa, Mohania, 2007). Note that the selected BJIs shall reduce the cost of executing no profitable queries generated by HPFIRST. This combination reduces the storage cost and especially the maintenance overhead. This approach shows the complementarity between horizontal partitioning and BJIs.

Vertical Partitioning and BJIs

Let A= $\{A_1, A_2, \ldots, A_K\}$ be the set of indexed attributes candidate for BJIs. For selecting **only one** BJI, the number of possible BJIs grows exponentially, and is given by:

$$\binom{K}{1} + \binom{K}{2} + \ldots + \binom{K}{K} = 2^K - 1$$

For (K=4), this number is 15. The total number of generating any combination of BJIs is given by:

$$\binom{2^K-1}{1} + \binom{2^K-1}{2} + \ldots + \binom{2^K-1}{2^K-1} = 2^{2^K-1} - 1$$

For K=4, **all possible BJIs** is $(2^{15} - 1)$. Therefore, the problem of efficiently finding the set of BJIs that minimize the total query processing cost while satisfying a storage constraint is very challenging to solve.

To vertically partition a table with m non primary keys, the number of possible fragments is equal to B(m), which is the m^{th} Bell number (Özsu & Valduriez, 1999). For a large values of m, $B(m) \cong m^m$. For example, for m = 10; B(10) ≅ 115 975. These values indicate that it is futile to attempt to obtain optimal solutions to the vertical partitioning problem. Many algorithms were proposed (Özsu & Valduriez, 1999) classified into two categories: (1) **Grouping:** starts by assigning each attribute to one fragment, and at each step, joins some of the fragments until some criteria is satisfied, (2) **Splitting:** starts with a table and decides on beneficial partitionings based on the frequency accesses of queries. The problem of generating BJIs is similar to the vertical partitioning problem. Since it tries to generate a group of attributes, where each group generates a BJI. Therefore, existing vertical partitioning algorithms could be adapted to BJIs selection.

Example

Let A= {Gender, City, Range} be three selection attributes candidate for indexing process. Suppose we use a vertical partitioning algorithm that generates two fragments {Gender} and {City, Range}. Based on these fragments, two BJIs (B1 and B2) can be generated as follows: (see Box 1).

FUTURE TRENDS

The selection of different optimization techniques is very challenging problem in designing and auto administrating very large databases. To get a better performance of queries, different optimization techniques should be combined, since each technique has its own advantages and drawbacks. For example, horizontal partitioning is an optimization structure that reduces query processing cost and storage and maintenance overhead. But it may generate a large number of fragments. BJIs may be used with horizontal partitioning to ovoid the explosion of fragments. Actually, only few algorithms are available for selecting bitmap join indexes. These algorithms are based on data mining techniques. It will be interested to adapt vertical partitioning algorithms for selecting bitmap join indexes and compare their performances with the existing ones. Another issue concerns the choice of threshold λ that identifies queries that get benefit from horizontal partitioning. This choice can be done using queries, selectivity factors of selection and join predicates, etc. This parameter may also play a role of tuning of the data warehouse.

CONCLUSION

Due to the complexity of queries on scientific and data warehouse applications, there is a need to develop query optimization techniques. We classify the existing techniques into two main categories: redundant and no redundant structures. We show a strong interdependency and complementarity between redundant structures and no redundant structures. For performance issue, we show how bitmap join indexes and data partitioning

Box 1.

CREATE BITMAP INDEX B1 ON SALES(CUSTOMER.Gender) FROM SALES S, CUSTOMER C WHERE S.CID= C.CID	CREATE BITMAP INDEX B2 ON SALES(CUSTOMER.City, Product. Range) FROM SALES S, CUSTOMER C, PRODUCT P WHERE S.CID= C.CID AND S.PID=P.PID

can interact each other. For selection issue, vertical partitioning can be easily dapted for selecting BJIs to speed up frequently asked queries. By exploiting the similarities between these two structures, an approach for combining them has been presented.

REFERENCES

Bellatreche L. & Boukhalfa K. (2005). An Evolutionary Approach to Schema Partitioning Selection in a Data Warehouse, 7th International Conference on Data Warehousing and Knowledge Discovery, 115-125

Bellatreche L., Boukhalfa K. & Abdalla H. I. (2006). SAGA : A Combination of Genetic and Simulated Annealing Algorithms for Physical Data Warehouse Design. In *23rd British National Conference on Databases*.

Bellatreche, L., Schneider, M., Lorinquer, H. & Mohania, M. (2004). Bringing together partitioning, materialized views and indexes to optimize performance of relational data warehouses. *International Conference on Data Warehousing and Knowledge Discovery (DAWAK),* 15-25.

Bellatreche L., Boukhalfa K. & Mohania (2007). Pruning Search Space of Physical Database. In 18th International Conference on Database and Expert Systems, 479-488

Bellatreche, Missaoui, Necir & Drias (2007). Selection and Pruning Algorithms for Bitmap Index Selection Problem Using Data Mining, 9th International Conference on Data Warehousing and Knowledge Discovery, 221-230

Datta, A., Moon, B. & Thomas, H. M. (2000). A case for parallelism in data warehousing and olap, International Conference on Database and Expert Systems *Workshops*, 226-231.

Johnson, T. (1999). Performance measurements of compressed bitmap indices. Processings of the International Conference on Very Large Databases, pp. 278–289.

Golfarelli, M., Rizzi, E. & Saltarelli, S. (2002). Index selection for data warehousing, International Workshop on Design and Management of Data Warehouses, 33–42.

Özsu. M. T. & Valduriez P. (1999). Principles of Distributed Database Systems: Second Edition. Prentice Hall.

Papadomanolakis, S. & Ailamaki A. (2004). Autopart: Automating schema design for large scientific databases using data partitioning. Proceedings of the 16th International Conference on Scientific and Statistical Database Management, 383–392.

Sanjay, A., Chaudhuri, S. & Narasayya, V. R. (2000). Automated selection of materialized views and indexes in Microsoft SQL server. *in Proceedings of 26th International Conference on Very Large Data Bases (VLDB'2000),* 496-505.

Sanjay A., Narasayya V. R. & Yang B. (2004). Integrating vertical and horizontal partitioning into automated physical database design. Proceedings of the ACM International Conference on Management of Data (SIGMOD), 359–370.

Stöhr T., Märtens H. & Rahm E.(2000). Multi-dimensional database allocation for parallel data warehouses. Proceedings of the International Conference on Very Large Databases, 273–284.

KEY TERMS

Bitmap Join Index: It is an index, where the indexed values come from one table, but the bitmaps point to another table.

Database Tuning: Is the activity of making a database application run quicker.

Fragmentation Attribute: Is an attribute of a dimension table participating in the fragmentation process of that table.

Horizontal Partitioning: Of a table R produces fragments R1, …, Rr, each of which contains a subset of tuples of the relation.

Indexed Attribute: Is an attribute of a dimension table participating in the definition of a BJI.

Maintenance Overhead: The cost required for maintaining any redundant structure.

Query Optimization: Is the process of reducing the query processing cost.

Selection Predicate: Selection predicate is the parameter of the selection operator of the relational algebra. It has the following form: Attribute θ Value; where Attribute is a column of a given table, $\theta \in \{=, <, >, \geq, \leq\}$, and Value $\in$ Domain(A_i)).

Vertical Partitioning: Of a table R produces fragments R1, …, Rr, each of which contains a subset of R's attributes as well as the primary key.

ENDNOTES

1. A dimension table is table containing the data for one dimension within a star schema. The primary key is used to link to the fact table, and each level in the dimension has a corresponding field in the dimension table.
2. A fact table is a central table in a star schema, containing the basic facts or measures of interest. Dimension fields are also included (as foreign keys) to link to each dimension table
3. This example serves as a running example along this paper.

Bridging Taxonomic Semantics to Accurate Hierarchical Classification

Lei Tang
Arizona State University, USA

Huan Liu
Arizona State University, USA

Jiangping Zhang
The MITRE Corporation, USA

INTRODUCTION

The unregulated and open nature of the Internet and the explosive growth of the Web create a pressing need to provide various services for content categorization. The hierarchical classification attempts to achieve both accurate classification and increased comprehensibility. It has also been shown in literature that hierarchical models outperform flat models in training efficiency, classification efficiency, and classification accuracy (Koller & Sahami, 1997; McCallum, Rosenfeld, Mitchell & Ng, 1998; Ruiz & Srinivasan ,1999; Dumais & Chen, 2000; Yang, Zhang & Kisiel, 2003; Cai & Hofmann, 2004; Liu, Yang, Wan, Zeng, Cheng & Ma, 2005). However, the quality of the taxonomy attracted little attention in past works. Actually, different taxonomies can result in differences in classification. So the quality of the taxonomy should be considered for real-world classifications. Even a semantically sound taxonomy does not necessarily lead to the intended classification performance (Tang, Zhang & Liu 2006). Therefore, it is desirable to construct or modify a hierarchy to better suit the hierarchical content classification task.

BACKGROUND

Hierarchical models rely on certain predefined content taxonomies. Content taxonomies are usually created for ease of content management or access, so semantically similar categories are grouped into a parent category. Usually, a subject expert or librarian is employed to organize the category labels into a hierarchy using some ontology information. However, such a taxonomy is often generated independent of data (e.g., documents). Hence, there may exist some inconsistency between the given taxonomy and data, leading to poor classification performance.

First, semantically similar categories may not be similar in lexical terms. Most content categorization algorithms are statistical algorithms based on the occurrences of lexical terms in content. Hence, a semantically sound hierarchy does not necessarily lead to the intended categorization result.

Second, even for the same set of categories, there could be different semantically sound taxonomies. Semantics does not guarantee a unique taxonomy. Different applications may need different category taxonomies. For example, sports teams may be grouped according to their locations such as Arizona, California, Oregon, etc and then the sports types such as football, basketball, etc.. Depending upon the application, they may also be grouped according to the sports types first and then locations. Both taxonomies are reasonable in terms of semantics. With a hierarchical classification model, however, the two taxonomies would likely result in different performances. Hence, we need to investigate the impact of different hierarchies (taxonomies) on classification.

In addition, semantics may change over time. For example, when the semantic taxonomy was first generated, people would not expect the category *Hurricane* related to *Politics*, and likely put it under *Geography*. However, after investigating the data recently collected, it is noticed that a good number of documents in category *Hurricane* are actually talking about the disasters Hurricane Katrina and Rita in the United States and the responsibility and the faults of FEMA during the crises. Based on the content, it is more reasonable to put

Hurricane under *Politics* for better classification. This example demonstrates the stagnant nature of taxonomy and the dynamic change of semantics reflected in data. It also motivates the data-driven adaptation of a given taxonomy in hierarchical classification.

MAIN FOCUS

In practice, semantics based taxonomies are always exploited for hierarchical classification. As the taxonomic semantics might not be compatible with specific data and applications and can be ambiguous in certain cases, the semantic taxonomy might lead hierarchical classifications astray. There are mainly two directions to obtain a taxonomy from which a good hierarchical model can be derived: *taxonomy generation via clustering* or *taxonomy adaptation via classification learning*.

Taxonomy Generation via Clustering

Some researchers propose to generate taxonomies from data for document management or classification. Note that the taxonomy generated here focus more on comprehensibility and accurate classification, rather than efficient storage and retrieval. Therefore, we omit the tree-type based index structures for high-dimensional data like R*-tree (Beckmann, Kriegel, Schneider & Seeger 1990), TV-tree (Lin, Jagadish & Faloutsos 1994), etc. Some researchers try to build a taxonomy with the aid of human experts (Zhang, Liu, Pan & Yang 2004, Gates, Teiken & Cheng 2005) whereas other works exploit some hierarchical clustering algorithms to automatically fulfill this task. Basically, there are two approaches for hierarchical clustering: *agglomerative* and *divisive*.

In Aggarwal, Gates & Yu (1999), Chuang & Chien (2004) and Li & Zhu (2005), all employ a hierarchical *agglomerative* clustering (HAC) approach. In Aggarwal, Gates & Yu (1999), the centroids of each class are used as the initial seeds and then projected clustering method is applied to build the hierarchy. During the process, a cluster with few documents is discarded. Thus, the taxonomy generated by this method may have different categories than predefined. The authors evaluated their generated taxonomies by some user study and found its perfomrance is comparable to the Yahoo directory. In Li & Zhu (2005), a linear discriminant projection is applied to the data first and then a hierarchical clustering method UPGMA (Jain & Dubes 1988) is exploited to generate a dendrogram which is a binary tree. For classification, the authors change the dendrogram to a two-level tree according to the cluster coherence, and hierarchical models yield classification improvement over flat models. But it is not sufficiently justified why a two-level tree should be adopted. Meanwhile, a similar approach, HAC+P was proposed by Chuang & Chien (2004). This approach adds one post-processing step to automatically change the binary tree obtained from HAC, to a wide tree with multiple children. However, in this process, some parameters have to be specified as the maximum depth of the tree, the minimum size of a cluster, and the cluster number preference at each level. These parameters make this approach rather ad hoc.

Comparatively, the work in Punera, Rajan & Ghosh (2005) falls into the category of *divisive* hierarchical clustering. The authors generate a taxonomy in which each node is associated with a list of categories. Each leaf node has only one category. This algorithm basically uses the centroids of the two most distant categories as the initial seeds and then applies Spherical K-Means (Dhillon, Mallela & Kumar, 2001) with k=2 to divide the cluster into 2 sub-clusters. Each category is assigned to one sub-cluster if majority of its documents belong to the sub-cluster (its ratio exceeds a predefined parameter). Otherwise, this category is associated to both sub-clusters. Another difference of this method from other HAC methods is that it generates a taxonomy with one category possibly occurring in multiple leaf nodes.

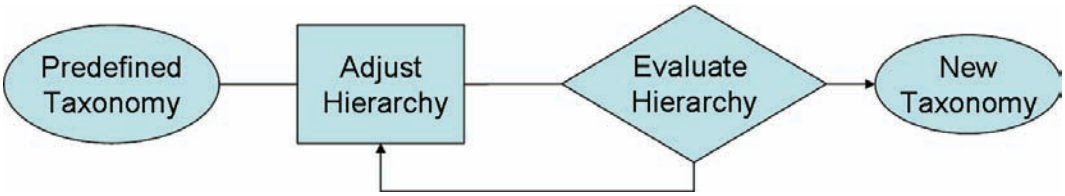
Taxonomy Adaptation via Classification Learning

Taxonomy clustering approach is appropriate if no taxonomy is provided at the initial stage. However, in reality, a human-provided semantic taxonomy is almost always available. Rather than "start from scratch", Tang, Zhang & Liu (2006) proposes to adapt the predefined taxonomy according the classification result on the data.

Three elementary hierarchy adjusting operations are defined:

- **Promote:** Roll up one node to upper level;
- **Demote:** Push down one node to its sibling;

Figure 1.



- **Merge:** Merge two sibling nodes to form a super node; Then a wrapper approach is exploited as in seen Figure 1.

The basic idea is, given a predefined taxonomy and training data, a different hierarchy can be obtained by performing the three elementary operations. Then, the newly generated hierarchy is evaluated on some validation data. If the change results in a performance improvement, we keep the change; otherwise, new change to the original taxonomy is explored. Finally, if no more change can lead to performance improvement, we output the new taxonomy which acclimatizes the taxonomic semantics according to the data.

In (Tang, Zhang & Liu 2006), the hierarchy adjustment follows a top-down traversal of the hierarchy. In the first iteration, only promoting is exploited to adjust the hierarchy whereas in the next iteration, demoting and merging are employed. This pair-wise iteration keeps running until no performance improvement is observed on the training data. As shown in their experiment, two iterations are often sufficient to achieve a robust taxonomy for classification which outperforms the predefined taxonomy and the taxonomy generated via clustering.

FUTURE TRENDS

Taxonomy can be considered as a form of prior knowledge. Adapting the prior knowledge to better suit the data is promising and desirable. Current works either abandon the hierarchy information or start taxonomy adaptation using a wrapper model. This short article provides some starting points that can hopefully lead to more effective and efficient methods to explore the prior knowledge in the future. When we better understand the problem of hierarchical classification and hierarchy consistency with data, we will investigate how to provide a filter approach which is more efficient to accomplish taxonomy adaptation.

This problem is naturally connected to Bayesian inference as well. The predefined hierarchy is the prior and the newly generated taxonomy is a "posterior" hierarchy. Integrating these two different fields—data mining and Bayesian inference, to reinforce the theory of taxonomy adaptation and to provide effective solution is a big challenge for data mining practitioners.

It is noticed that the number of features selected at each node can affect the performance and the structure of a hierarchy. When the class distribution is imbalanced, which is common in real-world applications, we should also pay attention to the problem of feature selection in order to avoid the bias associated with skewed class distribution (Forman, 2003; Tang & Liu, 2005). An effective criterion to select features can be explored in combination with the hierarchy information in this regard. Some general discussions and research issues of feature selection can be found in Liu & Motoda (1998) and Liu & Yu (2005).

CONCLUSION

Hierarchical models are effective for classification when we have a predefined semantically sound taxonomy. Since a given taxonomy may not necessarily lead to the best classification performance. Our task is how to obtain a data-driven hierarchy so that a reasonably good classifier can be inducted. In this article, we present an initial attempt to review and categorize the existing approaches: *taxonomy generation via clustering* and *taxonomy adaptation via classification learning*. It is anticipated that this active area of research will produce more effective and efficient approaches that are likely to emerge in a vast range of applications of web mining and text categorization.

REFERENCES

Aggarwal, C.C., Gates, S.C. & Yu, P.S. (1999). On the merits of building categorization systems by supervised clustering. *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 281-282).

Beckmann, N., Kriegel, H., Schneider, R. & Seeger, B.(1990) The R*-tree: an efficient and robust access method for points and rectangels. *Proceedings of the ACM SIGMOD international conference on management of data* (pp.322-331).

Cai, L. & Hofmann, T. (2004). Hierarchical document categorization with support vector machines, *Proceedings of the thirteenth ACM conference on Information and knowledge management* (pp. 78-87).

Chuang, S. & Chien, L. (2004). A practical web-based approach to generating topic hierarchy for text segments, *Proceedings of the thirteenth ACM conference on Information and knowledge management* (pp. 127-136)

Dhillon, I.S., Mallela, S. & Kumar, R. (2001) Efficient Clustering of Very Large Document Collections, in *Data Mining for Scientific and Engineering Applications*, Kluwer Academic.

Dumais, S. & Chen, H. (2000). Hierarchical classification of Web content. *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 256-263).

Forman, G. (2003). An Extensive Empirical Study of Feature Selection Metrics for Text Classification. *Journal of Machine Learning Research*, 3, 1289-1305.

Gates, S. C., Teiken, W., and Cheng, K. F. (2005). Taxonomies by the numbers: building high-performance taxonomies. In *Proceedings of the 14th ACM international Conference on information and Knowledge Management*. (pp. 568-577).

Jain, A.K.& Dubes, R.C. (Ed.). (1988). *Algorithms for clustering data.* Prentice-Hal Inc

Koller, D. & Sahami, M.(1997). Hierarchically classifying documents using very few words, *Proceedings of the Fourteenth International Conference on Machine Learning* (pp. 170-178).

Li, T. & Zhu, S. (2005). Hierarchical document classification using automatically generated hierarchy. *Proceedings of SIAM 2005 Data Mining Conference*(pp. 521-525).

Lin, K. I., Jagadish, H. V., and Faloutsos, C. 1994. The TV-tree: an index structure for high-dimensional data. *The VLDB Journal* 3, 4 (Oct. 1994), 517-542.

Liu, H. & Motoda, H.(Ed.). (1998). *Feature selection for knowledge discovery and data mining* Boston: Kluwer Academic Publishers.

Liu, H. & Yu, L. (2005). Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering*, 17, 491-502.

Liu, T., Yang, Y., Wan, H., Zeng, H., Chen, Z. & Ma, W. (2005). Support vector machines classification with a very large-scale taxonomy, *SIGKDD Explor. Newsl.*, 7(1), 36-43.

McCallum, A., Rosenfeld, R., Mitchell, T.M. & Ng, A.Y. (1998). Improving text classification by shrinkage in a hierarchy of classes, *Proceedings of the Fifteenth International Conference on Machine Learning* (pp. 359-367).

Punera, K., Rajan, S. & Ghosh, J. (2005). Automatically learning document taxonomies for hierarchical classification, *Special interest Tracks and Posters of the 14th international Conference on World Wide Web*(pp. 1010-1011).

Ruiz, M.E. & Srinivasan, P. (1999). Hierarchical neural networks for text categorization, *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 281-282).

Tang, L. & Liu, H. (2005) Bias analysis in text classification for highly skewed data, *Proceedings of the 5th IEEE international conference on Data Mining* (pp. 781-784).

Tang, L., Zhang, J. & Liu, H. (2006) *Acclimatizing taxonomic semantics for hierarchical content categorization Proceedings of the 12th Annual SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 384-393)

Yang, Y., Zhang, J. & Kisiel, B. (2003) A scalability analysis of classifiers in text categorization, *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 96-103).

Zhang, L., Liu, S., Pan, Y., and Yang, L. 2004. InfoAnalyzer: a computer-aided tool for building enterprise taxonomies. In *Proceedings of the Thirteenth ACM international Conference on information and Knowledge Management* (pp. 477-483).

KEY TERMS

Classification: A process of predicting the classes of unseen instances based on patterns learned from available instances with predefined classes.

Clustering: A process of grouping instances into clusters so that instances are similar to one another within a cluster but dissimilar to instances in other clusters.

Filter Model: A process that selects the best hierarchy without building hierarchical models.

Flat Model: A classifier outputs a classification from the input without any intermediate steps.

Hierarchical Model: A classifier outputs a classification using the taxonomy information in intermediate steps.

Hierarchy/Taxonomy: A tree with each node representing a category. Each leaf node represents a class label we are interested.

Taxonomy Adaptation: A process which adapts the predefined taxonomy based on the some data with class labels. All the class labels appear in the leaf node of the newly generated taxonomy.

Taxonomy Generation: A process which generates taxonomy based on some data with class labels so that all the labels appear in the leaf node of the taxonomy.

Wrapper Model: A process which builds a hierarchical model on training data and evaluates the model on validation data to select the best hierarchy. Usually, this process involves multiple constructions and evaluations of hierarchical models.

A Case Study of a Data Warehouse in the Finnish Police

Arla Juntunen
Department of Marketing and Management Helsinki School of Economics, Finland
Finland's Government Ministry of the Interior, Police Department, Finland

INTRODUCTION

The high level objectives of public authorities are to create value at minimal cost, and achieve ongoing support and commitment from its funding authority. Similar to the private sector, today's government agencies face a rapidly changing operating environment and many challenges. Where public organizations differ is that they need to manage this environment while answering to demands for increased service, reduced costs, fewer resources and at the same time increased efficiency and accountability. Public organization must cope with changing expectations of multiple contact groups, emerging regulation, changes in politics, decentralization of organization, and centralization of certain functions providing similar services, and growing demand for better accountability. The aim of public management is to create public value.

Public sector managers create value through their organization's performance and demonstrated accomplishments. The public value is difficult to define: it is something that exists within each community. It is created and affected by the citizens, businesses and organizations of that community (cf. also Moore, 1995). This increased interest to questions of value is partly due to the adoption of values and value-related concepts taken from business, like value creation and added value. It is argued that the public sector adopts business-like techniques to increase efficiency (Khademian, 1995; cf. Turban et al. 2007; Chen et al. 2005). In addition, there is a growing concern to the non-tangible, political, and ethical aspects of the public sector governance and actions (See Berg, 2001) Decision making that turns the resources in to public value is a daily challenge in the government (Khademian, 1999; Flynn, 2007) and not only because of the social or political factors. Most of decision problems are no longer well-structured problems that are easy to be solved by experience. Even problems that used to be fairly simple to define and solve are now much more complex because of the globalization of the economy, and rapid pace of changes in the technology and political and social environment. Therefore, modern decision makers often need to integrate quickly and reliably knowledge from different areas of data sources to use it in their decision making process. Moreover, the tools and applications developed for knowledge representations in key application areas are extremely diversified, therefore knowledge and data modeling and integration is important (See also the decision support systems (DSS) modeling methods and paradigms: Ruan et al., 2001; Carlsson & Fuller, 2002; Fink, 2002; Makowski & Wierzbicki, 2003). The application s of real-world problems and the abundance of different software tools allow to integrate several methods, specifications and analysis and to apply them to new, arising, complex problems.

In addition, business like methods and measurements to assist in managing and evaluating performance are hot topics in the government, and therefore, many government bodies are currently developing or have an existing data warehouse and a reporting solution. Recently, there has been a growing interest in measuring performance and productivity as a consequence of the convergence of two issues: (1) increased demand for accountability on the part of governing bodies, the media, and the public in general, and (2) a commitment of managers and government bodies to focus on results and to work more intentionally towards efficiency and improved performance (Poister, 2003).

This chapter discusses the issues and challenges of the adoption, implementation, effects and outcomes of the data warehouse and its use in analyzing the different operations of the police work, and the increased efficiency and productivity in the Finnish police organization due to the implementation and utilization of the data warehouse and reporting solution called PolStat (Police Statistics). Furthermore, the design of a data warehouse and analyzing system is not an easy task. It requires considering all phases from the requirements

specification to the final implementation including the ETL process. It should also take into account that the inclusion of different data items depends on both, users' needs and data availability in source systems. (Malinowski & Zimányi, 2007). The efficiency and productivity of the organization is measured with the key indicators defined and described by the police organization itself. The indicator data is stored in the data warehouse. The different organizational units are allowed to add their own static or dynamic reports into PolStat. The suggestions of how to further develop the content and measures are gathered from all over the organization as well as from external partners, like the customs and the boarder control.

The Data Warehouse and business intelligence are seen as a strategic management tool in the Finnish Police; a tool for better and comprehensive assessment of the police performance; a tool for balancing the most important perspectives of the police performance; and finally, a tool to communicate the current status of the police organization versus the development compared to previous years. Effectively, the Data Warehouse is also the means by which the police work as a whole will be made open and accountable to all interested parties. The system developed identifies what is to be achieved (target i.e. commonly agreed views of the police work and organization based on relevant measures), how to achieve the target, and by who, and the monthly estimates based on current data versus the previous years since 1997.

This chapter offers an overview of the development process of the data warehouse in the Finnish police, its critical success factors and challenges. The writer had an opportunity to make a follow-up study from the beginning of the process, and this contribution is an outline of the project initiated in the Police Department in 2002 and of main findings.

BACKGROUND

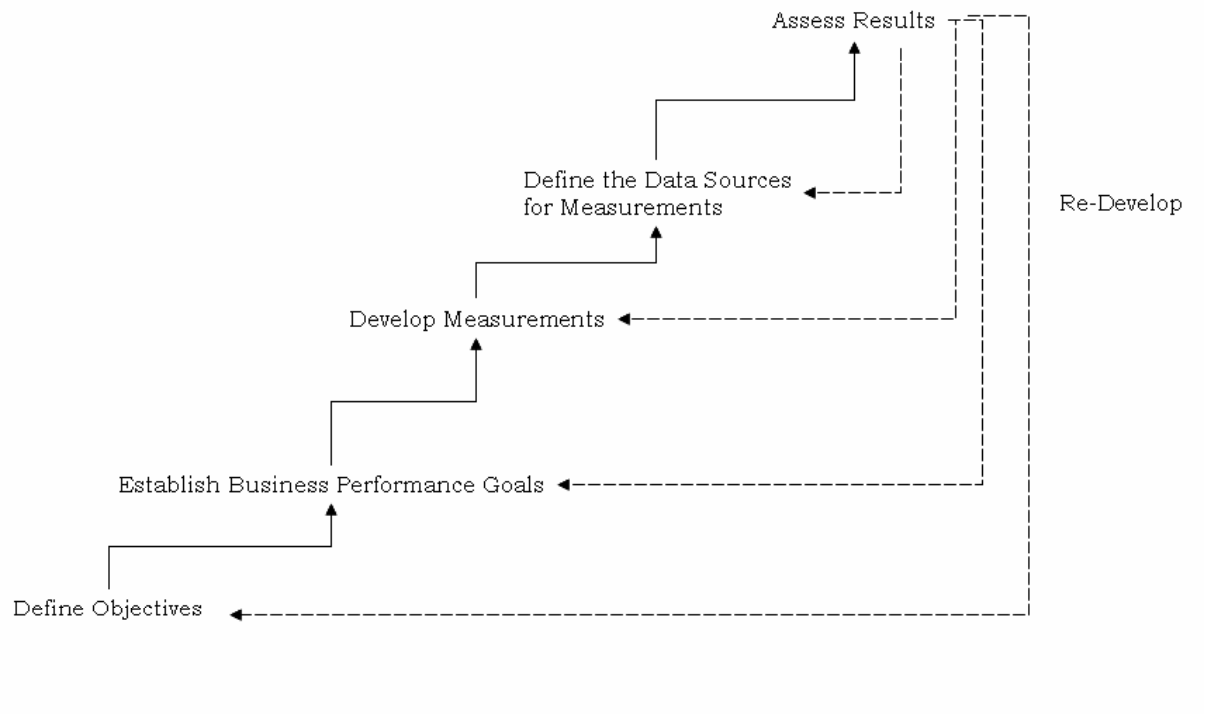
Storing (1981) stated that the separation of powers includes the assumption that certain kinds of functions are best performed in unique ways by distinctive organizational units: "With each branch distinct from the others, each can perform its function in its appropriate way." (Storing,1981, 59 - 60). The Finnish Police operates under the direction of the Ministry of the Interior. The Police Department of the Ministry of the Interior acts as the Supreme Police Command of Finland. The Supreme Police Command, which heads all police operations, decides on national strategies and ensures that the basic preconditions for effective operations exist. Within its command are five Provincial Police Commands; three national units, namely the National Bureau of Investigation, the Security Police and the National Traffic Police, the Police IT Management Agency, the Police Technical Center and the Helsinki Police Department and the police training establishments.

Inside the police organization, the challenge to improve the management by results–steering and –reporting has been among the main drivers why the development of the data warehouse started. The most important drivers to built a new data warehouse based on Oracle database, Informatica (ETL -extract, transform and load -tool), and Cognos Reporting and Analyzing tools in 2002 was the need to improve the strategic management and planning of the Finnish Police, and to develop a more suitable and comprehensive management reporting system (Management Information System, MIS) and to be able to collect and combine data from internal and external data sources.

The data warehouse content development is based on organizational needs and current governmental strategies and 'hot topics'. The strategy and vision drives the development. The assessment model (see Figure 1) supports the strategy and strategic goals and it assists the organization to understand the cyclic process of the data warehouse development. The performance planning consists of defining the organizational strategy including its vision and strategic objectives, defining and developing specific measurable goals. The current data collection and analysis processes are developed after the initial planning phase.

The measures developed are used to assess the organizational effectiveness, productivity, accountability and public value. Performance measures can be divided to direct and indirect, hard and soft measures (Bayley 1996, 47-48). Direct and hard performance measures are for example crime rates and criminal victimizations. Direct and soft measures are for example public confidence in police work, and satisfaction with police actions or feeling insecure in public places like streets at night time. Indirect and hard performance measures are for example number of police and police response times, arrests and a number of follow-up contacts with crime victims. Indirect soft measures are for example the awareness of the public reputation of the police, and

Figure 1. Strategy and vision drives the assessment model. The model supports the strategy and strategic goals



how well police knows its operational environment in different areas of the country. Also, Bailey states that only direct measures show police effectiveness. Soft indicators are significant complement to hard measurements. They contribute to the public value and quality of work. They also affect the police behavior.

MAIN FOCUS

The creation of a corporate-wide data warehouse, particularly on a public sector, is a major undertaking. It involves several issues, like for example the public acquisition procedures, project management, development or acquisition of tools for user access, database maintenance, data transfer and quality issues (Radding, 1995).

The Finnish Police started its first reporting systems back in 1992 (Törmänen, 2003) with SAS Institute's reporting tools. The first data warehouse was ready in 1996, and the first data was gathered to a SAS database in 1997. The long initial development period from 1992 to 1996 was due to the lack of skilled resources and unclear goals. However, even in 1992 there was an explicit opinion within the management of the Finnish Police that an executive information system was needed with static monthly and yearly reports (Pike, 1992). Though, it was not realized then that the development of a reporting system would eventually lead to a data warehouse with multidimensional cubes and different data analyzing and mining possibilities.

The rapid changes in the operating environment, organization and new technologies led to the redefinition of the technology architecture and the creation of PolStat data warehouse in 2002-2004. The development process consisted of the conversion of the old SAS-database to an Oracle 7 database, and after that started the new content development process of each of the topic areas (see figure 2).The end-users and managers were actively involved in the conversion and the development process of the new data warehouse, OLAP-cubes and the static and dynamic reports.

The Figure 2 shows the general configuration of the PolStat data warehouse and reporting solution. It enables the organization to consolidate its data from multiple internal and external data sources. The internal data bases include information, for example of the following topics: citizens (customers), licenses (passports, driver's

licenses, gun permits), different types of crimes, traffic information, book keeping and finance information, employee information. The consolidated data is used to monitor, plan, analyze and estimate the different operations of the police work. The data warehouse is built to be scalable and robust: there are currently about 12,000 users. The data warehouse model is based on a star schema. The data base itself is scalable up to terabytes. The data warehouse's metadata management allows to search, capture, use and publish metadata objects as dimensions, hierarchies, measures, and performance metrics and report layout objects.

The old legacy systems are based on various different technology architectures due to the long development history. The oldest systems are from 1983 and the newest from 2007, therefore, there is not only one technological solution for interfaces but several based on the various technologies used during the last 25 years. The data mining tool used is SPSS. Cognos 7 is used for reporting and analyzing the data in the data warehouse. Currently, there is an ongoing project to upgrade into the Cognos 8 version.

The Cognos analysis solution enables the users to search through large, complex data sets using drag-and-drop techniques on their own computer. It allows the drill down through increasing levels of detail, and view by different dimensions such as crime types per region or by the police department. It allows viewing and analyzing data relationships graphically. The end-users can easily drill down, slice and dice, rank, sort, forecast, and nest information to gain greater insight into the crime trends, causes to sudden changes compared to previous years, and effects on resource planning (see Figure 3).

The OLAP offers multidimensional views to the data and the analysts can analyze the data from different viewpoints. The different users are allowed to view the data and create new reports. There are only a few users with admin-rights. Most users only view the basic reports that are made ready monthly or every three months. The Cognos has also a common XML-based report format. The technical information of the data warehouse solution is as follows: The server operating system is HP-UX; The Web-Server is Microsoft IIS;

Figure 2. POLSTAT- Data warehouse stores data from different operational databases and external data sources. The information is used to monitor, plan, analyze and estimate. The management can use the data gathered to steer the organization towards its goals timely, efficiently and productively. (Figure mod. from BI-consultant Mika Naatula/Platon Finland's presentation)

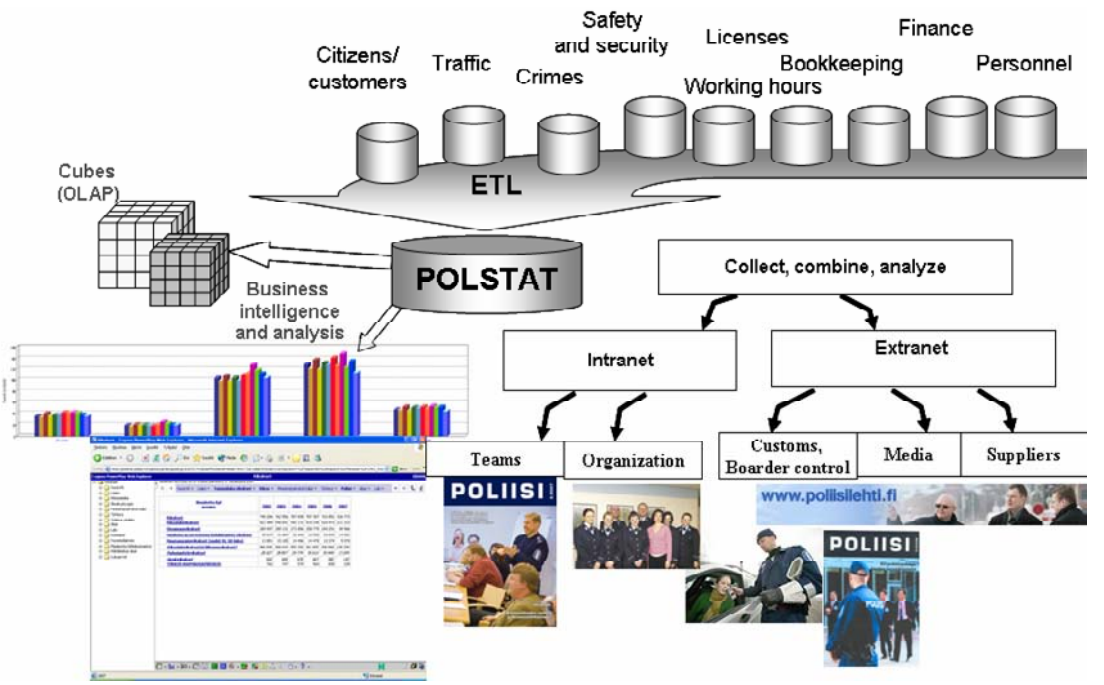
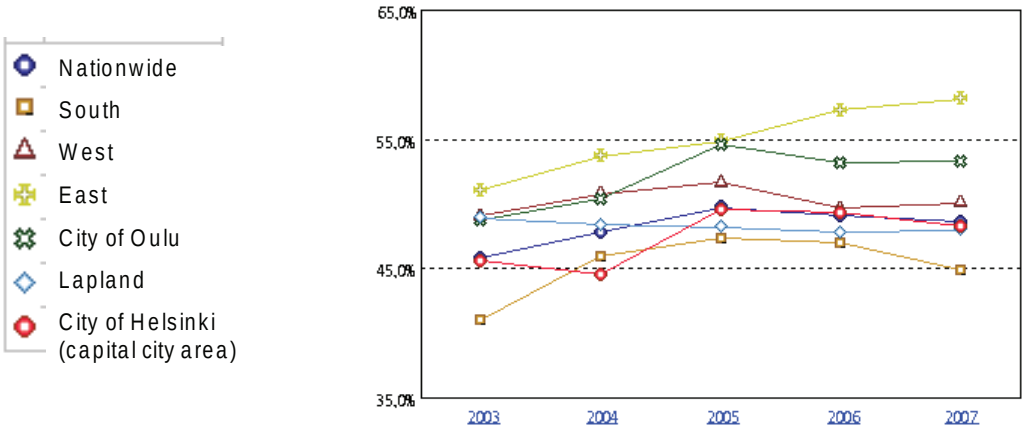


Figure 3. The percentage of solved crimes in jnauary-August 2003-2007 (source: Polstat)



Solved Crimes, % year	2003	2004	2005	2006	2007
Nationwide	45,8%	47,9%	49,8%	49,1%	48,6%
South	41,1%	46,0%	47,4%	47,0%	44,9%
West	49,1%	50,8%	51,7%	49,7%	50,1%
East	51,1%	53,8%	54,9%	57,3%	58,2%
City of Oulu	48,8%	50,4%	54,6%	53,2%	53,3%
Lapland	49,0%	48,4%	48,2%	47,9%	48,0%
City of Helsinki (capital city area)	45,6%	44,5%	49,6%	49,3%	48,3%

and the authentication providers used is Cognos Series 7 Namespace, however, that is changing with the new Cognos version 8 installation.

Enablers/Critical Success Factors

1. **Top-level commitment:** Commitment of the Ministry of the Interior and of the Supreme Command of the Police has been the most critical enabler to guarantee the success of the data warehouse development. This commitment has come not merely in terms of statements, but also in the commitment of development and maintenance resources.
2. **Use of technology:** The Government uses secure and standardized solutions and platforms in the development of Information Technology architecture and solutions. They try to foresee the future trends in technology and find out the most suitable solutions that would be cost-effective, secure and easy to use for the whole organization.
3. **Communication:** Communication performance is an essential part of the success of the data warehouse. However, it can also be the most difficult one to accomplish well. The successful communication needs: (i) the clarity of the goals; (ii) the flexibility—not rigidity—in decision process; and (iii) organizational culture that supports open discussion and communication between teams and different levels of organization (See also Pandey and Garnett, 2006)
4. **Supportive human resources:** The most valuable suppliers in this development have been TietoEnator Plc as a business intelligence consultant partner and Cognos as reporting and analyzing tool provider. However, the most important human resources are the organization's own resources. The end-users, like managers and analysts, make the data in the warehouse alive with their own knowledge and interpretations of the existing data.
5. **"Small is good":** Considering the high cost of data warehousing technology and the inherent risks of failure, the Finnish Police conclude that

the success is better when implementation begins with the creation of small data marts (OLAPs) for end-users (analysts) with considerable end-user computing and data analyzing experience.

The challenges of the data warehouse development process have caused time-related delays, re-prioritizing tasks and activities in the content development and maintenance.

Constraints/Challenges

1. **Resource instability:** Governmental and organizational changes as well as the current and time-related "hot topics" have an effect on the resource allocations. They can either reduce or increase the resource allocations. Lack of skilled staff to maintain and develop the data warehouse and reporting system has caused some delays in the content development.
2. **Data Security:** Data security can impact data warehousing projects and especially at the police even when there is no intent to make the data available outside the organization. This includes data transfer from an external data source and to the organization from external sources. Sometimes the data is protected by law and it is strictly defined who can access the data; sometimes data is protected by intra-departmental rules and by different user profiles with different data access.
3. **Technology Architecture Issues:** Like many public and private organizations, the police organization faces the problem of legacy systems that are several decades old and made with old and different technologies. These databases contain critically important data for daily operational management. Even more challenging are the constant maintenance of these old legacy systems due to changing operational environment, new crimes and laws. Also, old legacy systems are being renewed piece by piece. This creates more challenges in keeping the content of the data warehouse up to date.
4. **Changing organizational structures:** Changing organizational and reporting structures are challenging when the data sometimes needs to be seen both with the structure of the old organization and with the structure of the new organization.

FUTURE TRENDS

During the 1990s, a concept of Knowledge Management(See e.g. Davenport, 2001; Möller and Svahn, 2006; Nonaka et al., 2001) emerged, and both the public and private sector researchers accepted it in the information technology and management literature. Sharing, transferring and storing organizational knowledge has stressed the idea of further developing a data warehouse to support organizational learning and knowledge management within the organization (cf. Malinowski & Zimányi, 2007). The requirement specification and gathering can depend on the users, operational data sources or both. Therefore, the future challenges lies on designing and modeling the spatial and temporal data warehouses for various data requirements. Other challenges are to integrate data from diverse source systems in the face of organizational or economic constraints that require those systems to remain autonomous. Important future research challenges therefore include schema integration and evolution, and the query processing in such a heterogeneous environment (See March et al., 2000; March & Hevner, 2007; Phuboon-ob & Auepanwiriyakul, 2007).

The Finnish Police's plans include developing the data warehouse to support organizational learning because sharing new ideas, information and knowledge with the colleagues working in the same field are important in developing organizational resources and to be more effective and productive. Moreover, a substantial portion of the Finnish Police workforce will become eligible to retire within ten years. Workforce planning, organizational learning and knowledge management are integrated together because they are critical in ensuring that the different organizational units have sufficient and skilled staff to account for these retirements. Besides, high staff turnover, and lack of knowledge based on experience can hinder organizational effectiveness and productivity. In addition, to survive in the fast-changing environment the police as "an adaptive organization" (Radjou, et al., 2006) would have to be more like a shifting "constellation" (Mintzberg, 1998; Toffler, 1985) that has linkages (Pinfield et al., 1974) with its decentralized and semi-autonomous organizational units. These linkages can be for example linked organizational groups like functional teams, cross-functional teams, special task forces or project teams (Edgelow, 2006). The adaptive organization is a networked organization with internal

and external networks and flexible decision processes (Larraine, 2002; Möller et al. 2005, 2006).

CONCLUSION

This increased interest in questions of value creation and the never-ending struggle to develop passable result indicators and performance measures are probably due to the complexity of public service provision, and also because the different values and rationalities of public sector services are indeed incompatible with the private sector (See Berg, 2001). Still, this continuing discussion of public value and accountability of public sector actors, and how to measure and evaluate the performance and effectiveness means also increasing efforts to maintain the content of the data warehouse to support the new and growing demands.

REFERENCES

Bayley, David H. (1996). Measuring Overall Effectiveness. In Hoover (ed.) Q*uantifying Quality in Policing*. Police Executive Research Forum. Washington.

Berg, A. M. (2001). "The concept of value in public sector reform discourse", *Concept and Transformation*. Volume 6, Number 1, September 2001: 39-57(19)

Bovens, M. (2005). Public accountability, in: C. Pollit et al., *Oxford Handbook of Public Management*, Oxford: Oxford University Press.

Carlsson, C. & Fuller, R. (2002). *Fuzzy Reasoning in Decision Making and Optimization*, Physica Verlag, New York.

Chen, M., Zhang, D. & Zhou. L. (2005). Empowering Collaborative commerce with Web services enabled business process management systems. *Decision Support Systems,* Volume 43 , Issue 2 (March 2007): 530-546.

Davenport, Thomas H. (2001). *From Data to Knowledge*. San Jose, California, Oracle Inc.

Dodgson, M. (1993). "*Organizational learning: A review of some literatures*". *Organization Studies*, 14/3: 375-394.

Edgelow, C. (2006). *Helping Organizations Change*. Sundance Consulting Inc., February 2006.

Fink, E.:(2002). *Changes of Problem Representation*, Springer Verlag, Berlin, New York.

Flynn, N. (2007). *Public Sector Management*. 5th edition. Sage. UK.

Inmon, W. (1996). *Building the Data Warehouse*. John Wiley and Sons.

Khademian, A. (1995). Recent Changes in Public Sector Governance. Education Commission of the States, Denver. Co. [www.ecs.org]

Khademian, A. (1999). Reinventing a Government Corporation: Professional Priorities and a Clear Bottom Line . *Public Administration Review,* 55(1), 17– 28.

Kimball, R. (1996). *The Data Warehouse Toolkit*. John Wiley and Sons.

Larraine, S. (2002). Dynamic Leader Adaptive Organization: Ten Essential Traits for Managers. Wiley.

Makowski, M. & Wierzbicki, A. (2003). Modeling knowledge in global information networks, *4th Global Research Village Conference. Importance of ICT for Research an Science: Science Policies for Economies in Transition*,KBN (the Polish State Committee for Scientific Research), and OECD (the Organization for Economic Co-operation and Development), Information Processing Centre, Warsaw, pp. 173–186.

Malinowski, E. & Zimányi, E. (2007) *Designing Conventional, Spatial, and Temporal Data Warehouses: Concepts and Methodological Framework*. Thesis. Universite Libre de Bruxelles Faculte des Sciences Appliquees Service Ingenierie Informatique et de la Decision (CoDE). Belgium.

March, S. T., Hevner, A. & Ram, S. (2000).Research Commentary: An Agenda for Information Technology Research in Heterogeneous and Distributed Environments. *Information Systems Research*, Vol. 11, No. 4, December 2000, pp. 327-341.

March, S. T. & Hevner, A. (2007). Integrated decision support systems: A data warehousing perspective. *Decision Support Systems.* Volume 43, Issue 3. April 2007: 1031-1043.

Mintzberg, H., Ahlstrand, B. & Lampe, J. (1998) Strategy Safari. The complete guide through the wilds of strategic management. Prentice Hall Financial Times, London.

Moore, M. H. (1995). *Creating Public Value: Strategic Management in Government*. Cambridge, MA: Harvard University Press.

Möller, K., Rajala, A. & Svahn, S. (2005). Strategic Business Nets—Their Types and Management *Journal of Business Research*, 1274-1284.

Möller, K. & Svahn, S. (2006). Role of Knowledge in Value Creation in Business Nets. *Journal of Management Studies,* Vol. 43, Number 5, July 2006: 985-1007.

Nonaka, I., Toyama, R. & Konno, N. (2001). SECI, Ba and leadership: a unified model of dynamic knowledge creation. In: Nonaka, I & Teece, D. J. (Eds.). *Managing industrial knowledge. Creation, transfer and utilization*. London, Sage: 13-43.

Pandey, S. K. & Garnett, J. L. (2006). "Exploring Public Sector Communication Performance: Testing a Model and Drawing Implications",*Public Administration Review*, January 2006: 37-51.

Phuboon-ob, J. & Auepanwiriyakul, R. (2007). Two-Phase Optimization for Selecting Materialized Views in a Data Warehouse. *Proceedings of World Academy of Science, Engineering and Technology.* Volume 21 January 2007**.** ISSN 1307-6884.

Pike (1992). Pike-projektin loppuraportti. Sisäasiainministeriö.Poliisiosaston julkaisu. Series A5/92.

Pinfield, L.T., Watzke, G.E., & Webb, E.J. (1974). "Confederacies and Brokers: Mediators Between Organizations and their Environments," in H.Leavitt, L. Pinfield & E. Webb (Eds.), *Organizations of the Future: Interaction with the External Environment*, Praeger, New York, 1974, 83-110.

Poister, T. H. (2003). *Measuring performance in public and nonprofit organizations*. San Francisco, CA: John Wiley & Sons, Inc.

Pollitt, C. (2003). *The essential public manager*, London: Open University Press/McGraw-Hill.

Pritchard, A. (2003). Understanding government output and productivity. *Economic Trends*, 596: 27-40.

Radding, A. (1995). "Support Decision Makers with a Data Warehouse," *Datamation,* Vol. 41, Number 5, March 15, 53-56.

Radjou, N., Daley E, Rasmussen M. & Lo, H. (2006). *"The Rise of Globally Adaptive Organizations.* The World Isn't Flat Till Global Firms Are Networked, Risk-Agile, And Socially Adept , *Forrester,* published in the Balancing Risks And Rewards In A Global Tech Economy -series.

Romzek, B.S. and M.J. Dubnick (1998). Accountability, in: J.M. Shafritz (ed.), *International encyclopaedia of public policy and administration*, volume 1, Boulder: Westview Press.

Ruan, D., Kacprzyk, J. & Fedrizzi, M. (Eds). (2001). *Soft Computing for Risk Evaluation and Management*, Physica Verlag, New York.

Toffler, A.(1985). *The Adaptive Corporation*, McGraw Hill, New York.

Törmänen, A. (2002). *Tietovarastoinnin kehitys poliisiorganisaatiossa 1992-2002.* Sisäasiainministeriö, poliisiosaston julkaisusarja Ministry of the Interior/ Police Department Publications, 11/2003. ISBN 951-734-513-5.

Tolentino, A. (2004). *New concepts of productivity and its improvement*. European Productivity Network Seminar. International Labour Office. Retrieved June, 2007, from http://www.ilo.org/dyn/empent/docs/F1715412206/New%20Concepts%20of%20Productivity.pdf

Turban, E.,Leidner, D., McLean, E. & Wetherbe, J. (2007). ***Information*** *Technology for* ***Management****: Transforming Organizations in the Digital Economy.* John Wiley & Sons, Inc. New York, NY, USA

KEY TERMS

Accountability: It is a social relationship in which an actor feels an obligation to explain and to justify his or her conduct to some significant other. It presupposes that an organization has a clear policy on who is accountable to whom and for what. It involves the expectation that the accountable group will face consequences of its actions and be willing to accept advice

or criticism and to modify its practices in the light of that advice or criticism.

Data Warehouse: A data warehouse is a relational database that is designed for query and analysis rather than for transaction processing. It usually contains historical data derived from transaction data, but it can include data from both internal and external sources. It separates analysis workload from transaction workload and enables an organization to consolidate its data from multiple sources. In addition to a relational database, a data warehouse environment includes an extraction, transportation, transformation, and loading (ETL) solution, an online analytical processing (OLAP) engine, client analysis tools, and other applications that manage the process of gathering data and delivering it to business users.

Effectiveness and Efficiency: In this chapter, **effectiveness** in the public sector means that the organization is able to achieve its goals and be competitive. **Efficiency** means that the resources are not wasted.

ETL: ETL-tool is an extract, transform and load -tool. ETL is also a process in data warehousing that involves extracting data from different sources, transforming it to fit organizational needs, and finally loading it into the data warehouse.

Knowledge Management: (KM) deals with concept of how organizations, groups, and individuals handle their knowledge in all forms, in order to improve organizational performance.

Online Analytical Processing (OLAP): OLAP is a category of software tools providing analysis of data stored in a database. OLAP tools enable users to analyze different dimensions of multidimensional data. For example, it provides time series and trend analysis views. [http://en.wikipedia.org/wiki/OLAP_cube]

Organizational Learning (OL): It is the way organizations "build, supplement, and organize knowledge and routines around their activities and within their cultures and adapt and develop organizational efficiency by improving the use of the broad skills of their workforces" (Dodgson 1993: 377)

Productivity: Public sector productivity involves efficiency and outputs, it also involves effectiveness and outcomes.

Classification and Regression Trees

Johannes Gehrke
Cornell University, USA

INTRODUCTION

It is the goal of classification and regression to build a data mining model that can be used for prediction. To construct such a model, we are given a set of training records, each having several attributes. These attributes can either be numerical (for example, age or salary) or categorical (for example, profession or gender). There is one distinguished attribute, the dependent attribute; the other attributes are called predictor attributes. If the dependent attribute is categorical, the problem is a classification problem. If the dependent attribute is numerical, the problem is a regression problem. It is the goal of classification and regression to construct a data mining model that predicts the (unknown) value for a record where the value of the dependent attribute is unknown. (We call such a record an unlabeled record.) Classification and regression have a wide range of applications, including scientific experiments, medical diagnosis, fraud detection, credit approval, and target marketing (Hand, 1997).

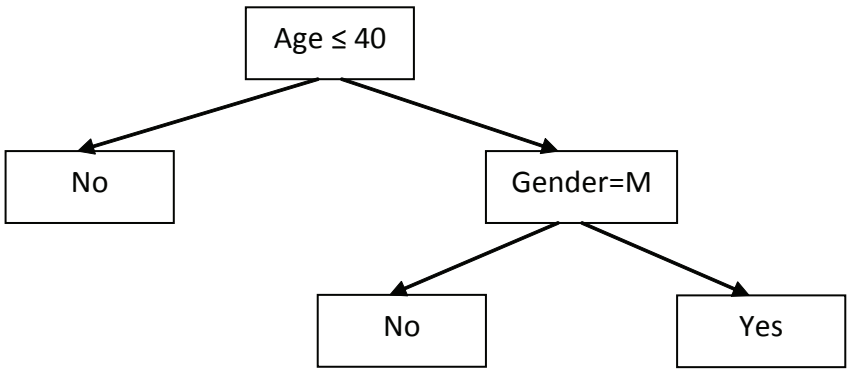
Many classification and regression models have been proposed in the literature, among the more popular models are neural networks, genetic algorithms, Bayesian methods, linear and log-linear models and other statistical methods, decision tables, and tree-structured models, the focus of this chapter (Breiman, Friedman, Olshen, & Stone, 1984). Tree-structured models, so-called decision trees, are easy to understand, they are non-parametric and thus do not rely on assumptions about the data distribution, and they have fast construction methods even for large training datasets (Lim, Loh, & Shih, 2000). Most data mining suites include tools for classification and regression tree construction (Goebel & Gruenwald, 1999).

BACKGROUND

Let us start by introducing decision trees. For the ease of explanation, we are going to focus on binary decision trees. In binary decision trees, each internal node has two children nodes. Each internal node is associated with a predicate, called the splitting predicate, which involves only the predictor attributes. Each leaf node is associated with a unique value for the dependent attribute. A decision encodes a data mining model as follows: For an unlabeled record, we start at the root node. If the record satisfies the predicate associated with the root node, we follow the tree to the left child of the root, and we go to the right child otherwise. We continue this pattern through a unique path from the root of the tree to a leaf node, where we predict the value of the dependent attribute associated with this leaf node. An example decision tree for a classification problem, a classification tree, is shown in Figure 1. Note that a decision tree automatically captures interactions between variables, but it only includes interactions that help in the prediction of the dependent attribute. For example, the rightmost leaf node in the example shown in Figure 1 is associated with the classification rule: "If (Age >= 40) and (Salary > 80k), then YES", as classification rule that involves an interaction between the two predictor attributes age and salary.

Decision trees can be mined automatically from a training database of records where the value of the dependent attribute is known: A decision tree construction algorithm selects which attribute(s) to involve in the splitting predicates and the algorithm decides also on the shape and depth of the tree (Murthy, 1998).

Figure 1. An example classification tree



MAIN THRUST

Let us discuss how decision trees are mined from a training database. A decision tree is usually constructed in two phases. In the first phase, the growth phase, an overly large and deep tree is constructed from the training data. In the second phase, the pruning phase, the final size of the tree is determined with the goal to minimize the expected mis-prediction error (Quinlan, 1993).

There are two problems that make decision tree construction a hard problem. First, construction of the "optimal" tree for several measure of optimality is an NP-hard problem. Thus all decision tree construction algorithms grow the tree top-down according to the following greedy heuristic: At the root node, the training database is examined and a splitting predicate is selected. Then the training database is partitioned according to the splitting predicate, and the same method is applied recursively at each child node. The second problem is that the training database is only a sample from a much larger population of records. The decision tree has to perform well on records drawn from the population, not on the training database. (For the records in the training database we already know the value of the dependent attribute.)

Three different algorithmic issues need to be addressed during the tree construction phase. The first issue is to devise a split selection algorithm such that the resulting tree models the underlying dependency relationship between the predictor attributes and the dependent attribute well. During split selection, we have to make two decisions. First, we need to decide which attribute we will select as splitting attribute. Second, given the splitting attribute, we have to decide on the actual splitting predicate. For a numerical attribute X, splitting predicates are usually of the form $X \leq c$, where c is a constant. For example, in the tree shown in Figure 1, the splitting predicate of the root node is of this form. For a categorical attribute X, splits are usually of the form X in C, where C is a set of values in the domain of X. For example, in the tree shown in Figure 1, the splitting predicate of the right child node of the root is of this form. There exist decision trees that have a larger class of possible splitting predicates, for example, there exist decision trees with linear combinations of numerical attribute values as splitting predicates for example $\Sigma a_i X_i + c \geq 0$, where i ranges over all attributes (Loh & Shih, 1997). Such splits, also called oblique splits, result in shorter trees, however, the resulting trees are no longer easy to interpret.

The second issue is to devise a pruning algorithm that selects the tree of the right size. If the tree is too large, then the tree models the training database too closely instead of modeling the underlying population. One possible choice of pruning a tree is to hold out part of the training set as a test set and to use the test set to estimate the misprediction error of trees of different size. We then simply select the tree that minimizes the misprediction error.

The third issue is to devise an algorithm for intelligent management of the training database in case the training database is very large (Ramakrishnan & Gehrke, 2002). This issue has only received attention in the last decade, but there exist now many algorithms that can construct decision trees over extremely large, disk-

resident training databases (Gehrke, Ramakrishnan, & Ganti, 2000; Shafer, Agrawal, & Mehta, 1996).

In most classification and regression scenarios, we also have costs associated with misclassifying a record, or with being far off in our prediction of a numerical dependent value. Existing decision tree algorithms can take costs into account, and they will bias the model towards minimizing the expected misprediction cost instead of the expected misclassification rate, or the expected difference between the predicted and true value of the dependent attribute.

FUTURE TRENDS

Recent developments have expanded the types of model that a decision tree can have in its leaf nodes. So far, we assumed that each leaf node just predicts a constant value for the dependent attribute. Recent work however, has shown how to construct decision trees with linear models in the leaf nodes (Dobra & Gehrke, 2002). Another recent development in the general area of data mining is the use of ensembles of models, and decision trees are a popular model for use as a base model in ensemble learning (Caruana, Niculescu-Mizil, Crew, & Ksikes, 2004). Another recent trend is the construction of data mining models of high-speed data streams, and there have been adaptations of decision tree construction algorithms to such environments (Domingos & Hulten, 2002). A last recent trend is to take adversarial behavior into account, for example in classifying spam. In this case, an adversary who produces the records to be classified actively changes her behavior over time to outsmart a static classifier (Dalvi, Domingos, Mausam, Sanghai, & Verma, 2004).

CONCLUSION

Decision trees are one of the most popular data mining models. Decision trees are important since they can result in powerful predictive models while at the same time they allow users to get insight into the phenomenon that is being modeled.

REFERENCES

Breiman, L., Friedman, J. H., Olshen, R. A. & Stone, C. J. (1984). Classification and Regression Trees, Kluwer Academic Publishers.

Caruana, R., Niculescu-Mizil, A., Crew, R, & Ksikes A. (2004). Ensemble selection from libraries of models. In Proceedings of the Twenty-first International Conference (ICML 2004), Banff, Alberta, Canada, July 4-8, 2004. ACM 2004

Dobra, A. & Gehrke, J. (2002). SECRET: A Scalable Linear Regression Tree Algorithm. In Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2002). Edmonton, Alberta, Canada.

Domingos, P. & Hulten G. (2002). Learning from Infinite Data in Finite Time. Advances in Neural Information Processing Systems 14, 673-680. Cambridge, MA: MIT Press.

Dalvi, N., Domingos, P., Mausam, Sanghai, S., & Verma, D. (2004). Adversarial Classification. In Proceedings of the Tenth International Conference on Knowledge Discovery and Data Mining, 99-108. Seattle, WA: ACM Press.

Gehrke, J., Ramakrishnan, R. & Ganti, V. (2000). 'Rainforest – A Framework For Fast Decision Tree Construction Of Large Datasets', Data Mining and Knowledge Discovery 4(2/3), 127-162.

Goebel, M. & Gruenwald, L. (1999). 'A survey of data mining software tools', SIGKDD Explorations 1(1), 20-33.

Hand, D. (1997), Construction and Assessment of Classification Rules, John Wiley & Sons, Chichester, England.

Lim, T.-S., Loh, W.-Y. & Shih, Y.-S. (2000). 'A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms', Machine Learning 48, 203-228.

Loh, W.-Y. & Shih, Y.-S. (1997). 'Split selection methods for classification trees', Statistica Sinica 7(4), 815-840.

Murthy, S. K. (1998). 'Automatic construction of decision trees from data: A multi-disciplinary survey', Data Mining and Knowledge Discovery 2(4), 345-389.

Quinlan, J. R. (1993). C4.5: Programs for Machine Learning, Morgan Kaufman.

Ramakrishnan, R. & Gehrke, J. (2002). Database Management Systems, Third Edition, McGrawHill.

Shafer, J., Agrawal, R. & Mehta, M. (1996). SPRINT: A scalable parallel classifier for data mining. In Proceedings of the 22nd International Conference on Very Large Databases, Bombay, India.

KEY TERMS

Attribute: Column of a dataset

Categorical Attribute: Attribute that takes values from a discrete domain.

Numerical Attribute: Attribute that takes values from a continuous domain.

Decision Tree: Tree-structured data mining model used for prediction, where internal nodes are labeled with predicates ("decisions") and leaf nodes are labeled with data mining models.

Classification Tree: A decision tree where the dependent attribute is categorical.

Regression Tree: A decision tree where the dependent attribute is numerical.

Splitting Predicate: Predicate at an internal node of the tree; it decides which branch a record traverses on its way from the root to a leaf node.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 141-143, copyright 2005 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

Classification Methods

Aijun An
York University, Canada

INTRODUCTION

Generally speaking, classification is the action of assigning an object to a category according to the characteristics of the object. In data mining, classification refers to the task of analyzing a set of pre-classified data objects to learn a model (or a function) that can be used to classify an unseen data object into one of several predefined classes. A data object, referred to as an example, is described by a set of attributes or variables. One of the attributes describes the class that an example belongs to and is thus called the class attribute or class variable. Other attributes are often called independent or predictor attributes (or variables). The set of examples used to learn the classification model is called the training data set. Tasks related to classification include regression, which builds a model from training data to predict numerical values, and clustering, which groups examples to form categories. Classification belongs to the category of supervised learning, distinguished from unsupervised learning. In supervised learning, the training data consists of pairs of input data (typically vectors), and desired outputs, while in unsupervised learning there is no a priori output.

Classification has various applications, such as learning from a patient database to diagnose a disease based on the symptoms of a patient, analyzing credit card transactions to identify fraudulent transactions, automatic recognition of letters or digits based on handwriting samples, and distinguishing highly active compounds from inactive ones based on the structures of compounds for drug discovery.

BACKGROUND

Classification has been studied in statistics and machine learning. In statistics, classification is also referred to as discrimination. Early work on classification focused on discriminant analysis, which constructs a set of discriminant functions, such as linear functions of the predictor variables, based on a set of training examples to discriminate among the groups defined by the class variable. Modern studies explore more flexible classes of models, such as providing an estimate of the join distribution of the features within each class (e.g. Bayesian classification), classifying an example based on distances in the feature space (e.g. the k-nearest neighbor method), and constructing a classification tree that classifies examples based on tests on one or more predictor variables (i.e., classification tree analysis).

In the field of machine learning, attention has more focused on generating classification expressions that are easily understood by humans. The most popular machine learning technique is decision tree learning, which learns the same tree structure as classification trees but uses different criteria during the learning process. The technique was developed in parallel with the classification tree analysis in statistics. Other machine learning techniques include classification rule learning, neural networks, Bayesian classification, instance-based learning, genetic algorithms, the rough set approach and support vector machines. These techniques mimic human reasoning in different aspects to provide insight into the learning process.

The data mining community inherits the classification techniques developed in statistics and machine learning, and applies them to various real world problems. Most statistical and machine learning algorithms are memory-based, in which the whole training data set is loaded into the main memory before learning starts. In data mining, much effort has been spent on scaling up the classification algorithms to deal with large data sets. There is also a new classification technique, called association-based classification, which is based on association rule learning.

MAIN THRUST

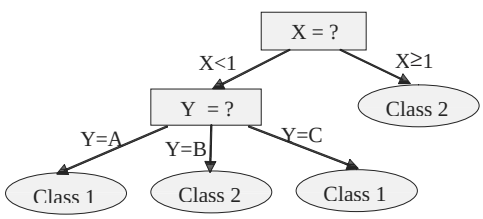
Major classification techniques are described below. The techniques differ in the learning mechanism and in the representation of the learned model.

Decision Tree Learning

Decision tree learning is one of the most popular classification algorithms. It induces a decision tree from data. A decision tree is a tree structured prediction model where each internal node denotes a test on an attribute, each outgoing branch represents an outcome of the test, and each leaf node is labeled with a class or class distribution. A simple decision tree is shown in Figure 1. With a decision tree, an object is classified by following a path from the root to a leaf, taking the edges corresponding to the values of the attributes in the object.

A typical decision tree learning algorithm adopts a top-down recursive divide-and-conquer strategy to construct a decision tree. Starting from a root node representing the whole training data, the data is split into two or more subsets based on the values of an attribute chosen according to a splitting criterion. For each subset a child node is created and the subset is associated with the child. The process is then separately repeated on the data in each of the child nodes, and so on, until a termination criterion is satisfied. Many decision tree learning algorithms exist. They differ mainly in attribute-selection criteria, such as information gain, gain ratio (Quinlan, 1993), gini index (Breiman, Friedman, Olshen, & Stone, 1984), etc., termination criteria and post-pruning strategies. Post-pruning is a technique that removes some branches of the tree after the tree is constructed to prevent the tree from over-fitting the training data. Representative decision tree algorithms include CART (Breiman et al., 1984) and C4.5 (Quinlan, 1993). There are also studies on fast and scalable construction of decision trees. Representative algorithms of such kind include RainForest (Gehrke, Ramakrishnan, & Ganti, 1998) and SPRINT (Shafer, Agrawal, & Mehta, 1996).

Figure 1. A decision tree with tests on attributes X and Y



Decision Rule Learning

Decision rules are a set of if-then rules. They are the most expressive and human readable representation of classification models (Mitchell, 1997). An example of decision rules is "if X<1 and Y=B, then the example belongs to Class 2". This type of rules is referred to as propositional rules. Rules can be generated by translating a decision tree into a set of rules – one rule for each leaf node in the tree. A second way to generate rules is to learn rules directly from the training data. There is a variety of rule induction algorithms. The algorithms induce rules by searching in a hypothesis space for a hypothesis that best matches the training data. The algorithms differ in the search method (e.g. general-to-specific, specific-to-general, or two-way search), the search heuristics that control the search, and the pruning method used. The most widespread approach to rule induction is *sequential covering*, in which a greedy general-to-specific search is conducted to learn a disjunctive set of conjunctive rules. It is called sequential covering because it sequentially learns a set of rules that together cover the set of positive examples for a class. Algorithms belonging to this category include CN2 (Clark & Boswell, 1991), RIPPER (Cohen, 1995) and ELEM2 (An & Cercone, 1998).

Naive Bayesian Classifier

The naive Bayesian classifier is based on Bayes' theorem. Suppose that there are m classes, $C_1, C_2, \ldots, C_m$. The classifier predicts an unseen example X as belonging to the class having the highest posterior probability conditioned on X. In other words, X is assigned to class C_i if and only if:

$$P(C_i|X) > P(C_j|X) \text{ for } 1 \le j \le m,\ j \ne i.$$

By Bayes' theorem, we have

$$P(C_i \mid X) = \frac{P(X \mid C_i)P(C_i)}{P(X)}.$$

As $P(X)$ is constant for all classes, only $P(X \mid C_i)P(C_i)$ needs to be maximized. Given a set of training data, $P(C_i)$ can be estimated by counting how often each class occurs in the training data. To reduce the computational expense in estimating $P(X|C_i)$ for all possible Xs, the classifier makes a naïve assumption that the attributes

used in describing X are conditionally independent of each other given the class of X. Thus, given the attribute values (x_1, x_2, ... x_n) that describe X, we have:

$$P(X \mid C_i) = \prod_{j=1}^{n} P(x_j \mid C_i).$$

The probabilities $P(x_1|C_i), P(x_2|C_i), \ldots, P(x_n|C_i)$ can be estimated from the training data.

The naïve Bayesian classifier is simple to use and efficient to learn. It requires only one scan of the training data. Despite the fact that the independence assumption is often violated in practice, naïve Bayes often competes well with more sophisticated classifiers. Recent theoretical analysis has shown why the naive Bayesian classifier is so robust (Domingos & Pazzani, 1997; Rish, 2001).

Bayesian Belief Networks

A Bayesian belief network, also known as Bayesian network and belief network, is a directed acyclic graph whose nodes represent variables and whose arcs represent dependence relations among the variables. If there is an arc from node *A* to another node *B*, then we say that *A* is a parent of *B* and *B* is a descendent of *A*. Each variable is conditionally independent of its nondescendents in the graph, given its parents. The variables may correspond to actual attributes given in the data or to "hidden variables" believed to form a relationship. A variable in the network can be selected as the class attribute. The classification process can return a probability distribution for the class attribute based on the network structure and some conditional probabilities estimated from the training data, which predicts the probability of each class.

The Bayesian network provides an intermediate approach between the naïve Bayesian classification and the Bayesian classification without any independence assumptions. It describes dependencies among attributes, but allows conditional independence among subsets of attributes.

The training of a belief network depends on the senario. If the network structure is known and the variables are observable, training the network only consists of estimating some conditional probabilities from the training data, which is straightforward. If the network structure is given and some of the variables are hidden, a method of gradient decent can be used to train the network (Russell, Binder, Koller, & Kanazawa, 1995). Algorithms also exist for learning the netword structure from training data given observable variables (Buntime, 1994; Cooper & Herskovits, 1992; Heckerman, Geiger, & Chickering, 1995).

The *k*-Nearest Neighbour Classifier

The *k*-nearest neighbour classifier classifies an unknown example to the most common class among its *k* nearest neighbors in the training data. It assumes all the examples correspond to points in a *n*-dimensional space. A neighbour is deemed nearest if it has the smallest distance, in the Euclidian sense, in the *n*-dimensional feature space. When $k = 1$, the unknown example is classified into the class of its closest neighbour in the training set. The *k*-nearest neighbour method stores all the training examples and postpones learning until a new example needs to be classified. This type of learning is called instance-based or lazy learning.

The *k*-nearest neighbour classifier is intuitive, easy to implement and effective in practice. It can construct a different approximation to the target function for each new example to be classified, which is advantageous when the target function is very complex, but can be discribed by a collection of less complex local approximations (Mitchell, 1997). However, its cost of classifying new examples can be high due to the fact that almost all the computation is done at the classification time. Some refinements to the k-nearest neighbor method include weighting the attributes in the distance computation and weighting the contribution of each of the *k* neighbors during classification according to their distance to the example to be classified.

Neural Networks

Neural networks, also referred to as *artificial neural networks*, are studied to simulate the human brain although brains are much more complex than any artificial neural network developed so far. A neural network is composed of a few layers of interconnected computing units (neurons or nodes). Each unit computes a simple function. The input of the units in one layer are the outputs of the units in the previous layer. Each connection between units is associated with a weight. Parallel computing can be performed among the units in each layer. The units in the first layer take input and are called the input units. The units in the last layer produces the output of the networks and are called

the output units. When the network is in operation, a value is applied to each input unit, which then passes its given value to the connections leading out from it, and on each connection the value is multiplied by the weight associated with that connection. Each unit in the next layer then receives a value which is the sum of the values produced by the connections leading into it, and in each unit a simple computation is performed on the value - a sigmoid function is typical. This process is then repeated, with the results being passed through subsequent layers of nodes until the output nodes are reached. Neural networks can be used for both regression and classification. To model a classification function, we can use one output unit per class. An example can be classified into the class corresponding to the output unit with the largest output value.

Neural networks differ in the way in which the neurons are connected, in the way the neurons process their input, and in the propogation and learning methods used (Nurnberger, Pedrycz, & Kruse, 2002). Learning a neural network is usually restricted to modifying the weights based on the training data; the structure of the initial network is usually left unchanged during the learning process. A typical network structure is the *multilayer feed-forward neural network*, in which none of the connections cycles back to a unit of a previous layer. The most widely used method for training a feed-forward neural network is backpropagation (Rumelhart, Hinton, & Williams, 1986).

Support Vector Machines

The support vector machine (SVM) is a recently developed technique for multidimensional function approximation. The objective of support vector machines is to determine a classifier or regression function which minimizes the empirical risk (that is, the training set error) and the confidence interval (which corresponds to the generalization or test set error) (Vapnik, 1998).

Given a set of N linearly separable training examples $S = \{\mathbf{x}_i \in R^n \mid i = 1,2,...,N\}$, where each example belongs to one of the two classes, represented by $y_i \in \{+1,-1\}$, the SVM learning method seeks the optimal hyperplane $\mathbf{w} \cdot \mathbf{x} + b = 0$, as the decision surface, which separates the positive and negative examples with the largest margin. The decision function for classifying linearly separable data is:

$$f(\mathbf{x}) = sign(\mathbf{w} \cdot \mathbf{x} + b),$$

where $\mathbf{w}$ and b are found from the training set by solving a constrained quadratic optimization problem. The final decision function is

$$f(\mathbf{x}) = sign\left(\sum_{i=1}^{N} \alpha_i y_i (\mathbf{x}_i \cdot \mathbf{x}) + b\right).$$

The function depends on the training examples or which α_i is non-zero. These examples are called support vectors. Often the number of support vectors is only a small fraction of the original dataset. The basic SVM formulation can be extended to the nonlinear case by using nonlinear kernels that map the input space to a high dimensional feature space. In this high dimensional feature space, linear classification can be performed. The SVM classifier has become very popular due to its high performances in practical applications such as text classification and pattern recognition.

FUTURE TRENDS

Classification is a major data mining task. As data mining becomes more popular, classification techniques are increasingly applied to provide decision support in business, biomedicine, financial analysis, telecommunications and so on. For example, there are recent applications of classification techniques to identify fraudulent usage of credit cards based on credit card transaction databases; and various classification techniques have been explored to identify highly active compounds for drug discovery. To better solve application-specific problems, there has been a trend toward the development of more application-specific data mining systems (Han & Kamber, 2001).

Traditional classification algorithms assume that the whole training data can fit into the main memory. As automatic data collection becomes a daily practice in many businesses, large volumes of data that exceed the memory capacity become available to the learning systems. Scalable classification algorithms become essential. Although some scalable algorithms for decision tree learning have been proposed, there is still a need to develop scalable and efficient algorithms for other types of classification techniques, such as decision rule learning.

Previously, the study of classification techniques focused on exploring various learning mechanisms to improve the classification accuracy on unseen examples. However, recent study on imbalanced data sets has shown that classification accuracy is not an appropriate measure to evaluate the classification performance when the data set is extremely unbalanced, in which almost all the examples belong to one or more, larger classes and far fewer examples belong to a smaller, usually more interesting class. Since many real world data sets are unbalanced, there has been a trend toward adjusting existing classification algorithms to better identify examples in the rare class.

Another issue that has become more and more important in data mining is privacy protection. As data mining tools are applied to large databases of personal records, privacy concerns are rising. Privacy-preserving data mining is currently one of the hottest research topics in data mining and will remain so in the near future.

CONCLUSION

Classification is a form of data analysis that extracts a model from data to classify future data. It has been studied in parallel in statistics and machine learning, and is currently a major technique in data mining with a broad application spectrum. Since many application problems can be formulated as a classification problem and the volume of the available data has become overwhelming, developing scalable, efficient, domain-specific, and privacy-preserving classification algorithms is essential.

REFERENCES

An, A., & Cercone, N. (1998). ELEM2: A learning system for more accurate classifications. *Proceedings of the 12th Canadian Conference on Artificial Intelligence* (pp. 426-441).

Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). *Classification and regression trees*. Wadsworth International Group.

Buntine, W.L. (1994). Operations for learning with graphical models. *Journal of Artificial Intelligence Research, 2*, 159-225.

Castillo, E., Gutiérrez, J.M., & Hadi, A.S. (1997). *Expert systems and probabilistic network models.* New York: Springer-Verlag.

Clark P., & Boswell, R. (1991). Rule induction with CN2: Some recent improvements. *Proceedings of the 5th European Working Session on Learning* (pp. 151-163).

Cohen, W.W. (1995). Fast effective rule induction. *Proceedings of the 11th International Conference on Machine Learning* (pp. 115-123), Morgan Kaufmann.

Cooper, G., & Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine Learning, 9*, 309-347.

Domingos, P., & Pazzani, M. (1997). On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning, 29*, 103-130.

Gehrke, J., Ramakrishnan, R., & Ganti, V. (1998). RainForest - A framework for fast decision tree construction of large datasets. *Proceedings of the 24th International Conference on Very Large Data Bases* (pp. 416-427).

Han, J., & Kamber, M. (2001). *Data mining — Concepts and techniques*. Morgan Kaufmann.

Heckerman, D., Geiger, D., & Chickering, D.M. (1995) Learning bayesian networks: The combination of knowledge and statistical data. *Machine Learning, 20*, 197-243.

Mitchell, T.M. (1997). *Machine learning*. McGraw-Hill.

Nurnberger, A., Pedrycz, W., & Kruse, R. (2002). Neural network approaches. In Klosgen & Zytkow (Eds.), *Handbook of data mining and knowledge discovery* (pp. 304-317). Oxford University Press.

Pearl, J. (1986). Fusion, propagation, and structuring in belief networks. *Artificial Intelligence, 29*(3), 241-288.

Quinlan, J.R. (1993). *C4.5: Programs for machine learning*. Morgan Kaufmann.

Rish, I. (2001). An empirical study of the naive Bayes classifier. *Proceedings of IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence.*

Rumelhart, D.E., Hinton, G.E., & Williams, R.J. (1986). Learning representations by back-propagating errors. *Nature, 323*, 533-536.

Russell, S., Binder, J., Koller, D., & Kanazawa, K. (1995). Local learning in probabilistic networks with hidden variables. *Proceedings of the 14th Joint International Conference on Artificial Intelligence*, *2* (pp. 1146-1152).

Shafer, J., Agrawal, R., & Mehta, M. (1996). SPRINT: A scalable parallel classifier for data mining. *Proceedings of the 22th International Conference on Very Large Data Bases* (pp. 544-555).

Vapnik, V. (1998). *Statistical learning theory*. New York: John Wiley & Sons.

KEY TERMS

Backpropagation: A neural network training algorithm for feedforward networks where the errors at the output layer are propagated back to the previous layer to update connection weights in learning. If the previous layer is not the input layer, then the errors at this hidden layer are propagated back to the layer before.

Disjunctive Set of Conjunctive Rules: A conjunctive rule is a propositional rule whose antecedent consists of a conjunction of attribute-value pairs. A disjunctive set of conjunctive rules consists of a set of conjunctive rules with the same consequent. It is called disjunctive because the rules in the set can be combined into a single disjunctive rule whose antecedent consists of a disjunction of conjunctions.

Generic Algorithm: An algorithm for optimizing a binary string based on an evolutionary mechanism that uses replication, deletion, and mutation operators carried out over many generations.

Information Gain: Given a set E of classified examples and a partition $P = \{E_1, ..., E_n\}$ of E, the information gain is defined as:

$$entropy(E) - \sum_{i=1}^{n} entropy(E_i) * \frac{|E_i|}{|E|},$$

where $|X|$ is the number of examples in X, and $entropy(X) = -\sum_{j=1}^{m} p_j \log_2(p_j)$ (assuming there are m classes in X and p_j denotes the probability of the jth class in X). Intuitively, the information gain measures the decrease of the weighted average impurity of the partitions E_1, ..., E_n, compared with the impurity of the complete set of examples E.

Machine Learning: The study of computer algorithms that develop new knowledge and improve its performance automatically through past experience.

Rough Set Data Analysis: A method for modeling uncertain information in data by forming lower and upper approximations of a class. It can be used to reduce the feature set and to generate decision rules.

Sigmod Function: A mathematical function defined by the formula

$$P(t) = \frac{1}{1 + e^{-t}}$$

Its name is due to the sigmoid shape of its graph. This function is also called the standard logistic function.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 144-149, copyright 2005 by IGI Publishing, formerly known as Idea Group Publishing (an imprint of IGI Global).

Classification of Graph Structures

Andrzej Dominik
Warsaw University of Technology, Poland

Zbigniew Walczak
Warsaw University of Technology, Poland

Jacek Wojciechowski
Warsaw University of Technology, Poland

INTRODUCTION

Classification is a classical and fundamental data mining (machine learning) task in which individual items (objects) are divided into groups (classes) based on their features (attributes). Classification problems have been deeply researched as they have a large variety of applications. They appear in different fields of science and industry and may be solved using different algorithms and techniques: e.g. neural networks, rough sets, fuzzy sets, decision trees, etc. These methods operate on various data representations. The most popular one is information system/decision table (e.g. Dominik, & Walczak, 2006) denoted by a table where rows represent objects, columns represent attributes and every cell holds a value of the given attribute for a particular object. Sometimes it is either very difficult and/or impractical to model a real life object (e.g. road map) or phenomenon (e.g. protein interactions) by a row in decision table (vector of features). In such a cases more complex data representations are required e.g. graphs, networks. A graph is basically a set of nodes (vertices) connected by either directed or undirected edges (links). Graphs are used to model and solve a wide variety of problems including classification. Recently a huge interest in the area of graph mining can be observed (e.g. Cook, & Holder, 2006). This field of science concentrates on investigating and discovering relevant information from data represented by graphs.

In this chapter, we present basic concepts, problems and methods connected with graph structures classification. We evaluate performance of the most popular and effective classifiers on two kinds of classification problems from different fields of science: computational chemistry, chemical informatics (chemical compounds classification) and information science (web documents classification).

BACKGROUND

There are numerous types of pattern that can be used to build classifiers. Three of these are frequent, common and contrast patterns.

Mining patterns in graph dataset which fulfill given conditions is a much more challenging task than mining patterns in decision tables (relational databases). The most computationally complex tasks are subgraph isomorphism (determining if one smaller graph is included in other larger graph) and isomorphism (testing whether any two graphs are isomorphic (really the same). The former is proved to be NP-complete while the complexity of the latter one is still not known. All the algorithms for solving the isomorphism problem present in the literature have an exponential time complexity in the worst case, but the existence of a polynomial solution has not yet been disproved. A universal exhaustive algorithm for both of these problems was proposed by Ullman (1976). It operates on the matrix representation of graphs and tries to find a proper permutation of nodes. The search space can be greatly reduced by using nodes invariants and iterative partitioning. Moreover multiple graph isomorphism problem (for a set of graphs determine which of them are isomorphic) can be efficiently solved with canonical labelling (Fortin, 1996). Canonical label is a unique representation (code) of a graph such that two isomorphic graphs have the same canonical label.

Another important issue is generating all non-isomorphic subgraphs of a given graph. The algorithm for generating DFS (Depth First Search) code can used to enumerate all subgraphs and reduce the number of required isomorphism checking. What is more it can be improved by introducing canonical labelling.

Contrast patterns are substructures that appear in one class of objects and do not appear in other classes

whereas common patterns appear in more than one class of objects. In data mining, patterns which uniquely identify certain class of objects are called jumping emerging patterns (JEP). Patterns common for different classes are called emerging patterns (EP). Concepts of jumping emerging patterns and emerging patterns have been deeply researched as a tool for classification purposes in databases (Kotagiri, & Bailey, 2003). They are reported to provide high classification accuracy results. Frequent pattern is a pattern which appears in samples of a given dataset more frequently than specified threshold. Agarwal and Srikant proposed an efficient algorithm for mining frequent itemsets in the transaction database called Apriori.

In graph mining contrast graph is a graph that is subgraph isomorphic to at least one graph from particular class of graphs and is not subgraph isomorphic to any graph from any other class of graphs. The concept of contrast graphs was studied by Ting and Bailey (2006). They proposed an algorithm (containing backtracking tree and hypergraph traversal algorithm) for mining all disconnected contrast graphs from dataset. Common graph is subgraph isomorphic to at least one graph in at least two different classes of graphs while frequent graph is subgraph isomorphic to at least as many graphs in a particular class of graphs as specified threshold (minimal support of a graph). Kuramochi and Karypis (2001) proposed an efficient (using canonical labelling and iterative partitioning) Apriori based algorithm for mining frequent graphs.

MAIN FOCUS

One of the most popular approaches for graph classification is based on SVM (Support Vector Machines). SVMs have good generalization properties (both theoretically and experimentally) and they operate well in high-dimensional datasets. Numerous different kernels were designed for this method (Swamidass, Chen, Bruand, Phung, Ralaivola & Baldi, 2005).

Another approach is based on k-NN (k-Nearest Neighbors) method. The most popular similarity measures for this method use concept of MCS (maximum common subgraph) (Markov, Last, & Kandel, 2006).

Deshpande, Kuramochi and Karypis (2003) proposed classification algorithms that decouples graph discovery process from the classification model construction. These methods use frequent topological and geometric graphs.

Recently a new algorithm called the CCPC (Contrast Common Patterns Classifier) was proposed by Dominik, Walczak, & Wojciechowski (2007). This algorithm uses concepts that were originally developed and introduced for data mining (jumping emerging patterns - JEP and emerging patterns - EP). The CCPC approach uses minimal (with respect to size and inclusion (non-isomorphic)) contrast and common connected graphs. Classification process is performed by aggregating supports (or other measure based on support) of contrast graphs. Common graphs play marginal role in classification and are only used for breaking ties.

Applications: Chemical Compounds Classification

Chemical molecules have various representations depending on their dimensions and features. Basic representations are: 1-dimensional strings expressed in SMILES language (language that unambiguously describe the structure of chemical molecules using short ASCII strings), 2-dimensional topological graphs and 3-dimensional geometrical structures containing coordinates of atoms. We are particularly interested in 2-dimensional graph representation in which atoms correspond to vertices and bonds to edges. Nodes are labelled with molecule symbols and links with bond multiplicities. These graphs are typically quite small (in terms of number of vertices and edges) and the average number of edges per vertex is usually slightly above 2.

Sample classification problems in the area of chemical informatics and computational chemistry include: detection/prediction of mutagenicity, toxicity and anti-cancer activity of chemical compounds for a given organism. There are two major approaches to classifying chemical compounds: quantitative structure-activity relationships (QSAR) and structural approach. The former one (King, Muggleton, Srinivasan, Sternberg, 1996) requires genuine chemical knowledge and concentrates on physico-chemical properties derived from compounds while the latter one (Deshpande, Kuramochi, & Karypis, 2003; Kozak, Kozak, & Stapor, 2007; Dominik, Walczak, & Wojciechowski, 2007) searches directly structure of the compound and discover significant substructures (e.g. contrast, common, frequent

pattern) which discriminate between different classes of structures.

We evaluated performance of most popular and effective classification methods on three publicly available datasets: Mutag, PTC and NCI (Datasets: Mutag, PTC, NCI, 2007). The Mutag dataset consists of 188 chemical compounds (aromatic and heteroaromatic nitro-compounds) along with the information indicating whether they have mutagenicity in *Salmonella typhimurium*. The Predictive Toxicology Challenge (PTC) dataset reports carcinogenicity of 417 compounds for female mice (FM), female rats (FR), male mice (MM) and male rats (MR). There are four independent problems (for each sex-animal pair) in this dataset. The National Cancer Institute (NCI) dataset provides screening results for about 70 000 compounds that suppress or inhibit the growth of a panel of 73 human tumor cell lines corresponding to the concentration parameter GI50 (the concentration that causes 50% growth inhibition). Each cell line contains approximately 3500 compounds and information on their cancer-inhibiting action. For our research we choose 60 out of 73 cell lines. There are two decision classes in all of these three problems: positive and negative.

Table 1 reports classification accuracy for different datasets and algorithms (Dominik, Walczak, & Wojciechowski, 2007). We made a selection of best available (in case of algorithms parameters) results for the following methods: PD (Pattern Discovery (De Readt, & Kramer, 2001)), SVM (Support Vector Machine with different kernels (Swamidass, Chen, Bruand, Phung, Ralaivola & Baldi, 2005)) and the CCPC. For the Mutag and PTC datasets, classification accuracy was estimated through leave-one-out cross-validation. In case of NCI dataset cross-validation using 20 random 80/20 training/test splits was used. For this dataset reported values are average values from all 60 cell lines. Best results for each dataset are in bold face and second best are in italic face. NA indicates that the given value was not available. For all datasets the CCPC classifier outperformed other approaches in case of accuracy.

Applications: Web Documents Classification

There are three basic document representations: vector, graph and hybrid which is a combination of two previous ones. In vector model each term in a document becomes a feature (dimension). The value of each dimension in a vector is some measure based on frequency of appropriate term in a given document. In graph model terms refer to nodes. Nodes are connected by edges which provides both text structure (e.g. the order in which the words appear or the location of a word within the document) and document structure (e.g. markup elements (tags) inside HTML web document) information.

HTML web documents are often represented using standard model. According to this model web document is represented as a graph containing N labelled nodes corresponding to N unique most frequently appearing words in a document. Nodes are labelled with words they represent. Two nodes are connected by undirected labelled edge if they are adjacent in a document. An edge's label depends on the section in which two particular words are adjacent. There are three major sections: title, which contains the text related to the document's title and any provided keywords; link, which is text appearing in clickable hyperlinks on the document; and text, which comprises any of the readable text in the document (this includes link text but not title and keyword text).

One of the most popular approaches for document classification is based on k-NN (k-Nearest Neighbors) method. Different similarity measures were proposed

Table 1. Comparison of classification accuracy for chemical compounds

Algorithm	Mutag	PTC-FM	PTC-FR	PTC-MM	PTC-MR	NCI
PD	89.9	61.0	66.7	61.0	62.8	NA
SVM, 1D SMILES	85.6	63.0	*67.0*	*66.4*	57.6	NA
SVM, 2D Tanimoto	*90.4*	64.2	66.7	*66.4*	63.7	71.5
SVM, 2D MinMax	**91.5**	*64.5*	66.4	64.0	*64.5*	*72.3*
CCPC	**91.5**	**81.4**	**82.3**	**77.7**	**77.0**	**86.2**

for different document representations. For vector representation the most popular is cosine measure while for graph representation distance based on maximum common subgraph (MCS) is widely used (Markov, Last, & Kandel, 2006). Recently methods based on hybrid document representations have become very popular. They are reported to provide better results than methods using simple representations. Markov and Last (2005) proposed an algorithm that uses hybrid representation. It extracts subgraphs from a graph that represents document, then creates vector with Boolean values indicating relevant subgraphs.

In order to evaluate the performance of the CCPC classifier we performed several experiments on three publicly available, benchmark collections of web documents, called F-series, J-series and K-series (Datasets: PDDPdata, 2007). Each collection contains HTML documents which were originally news pages hosted at Yahoo (www.yahoo.com). Each document in every collection has a category (class) associated to the content of the document. The F-series collection contains 98 documents divided into 4 major categories, the J-series collection contains 185 documents assigned to 10 categories while the K-series consists of 2340 documents belonging to 20 categories. In all of those datasets document is assigned to exactly one category.

Table 2 reports classification accuracy for different datasets, document representations and algorithms (Dominik, Walczak, & Wojciechowski, 2007). We made a selection of best available (in case of algorithms parameters) results for following methods: k-NN (k-Nearest Neighbor) (Markov, Last, & Kandel, 2006; Markov, & Last, 2006) and the CCPC. For all document collections classification accuracy was estimated through leave-one-out cross-validation. Best results for each dataset are in bold face and second best are in italic face. The results show that our algorithm is competitive to existing schemes in terms of accuracy and data complexity (number of terms in document model used for classification).

FUTURE TRENDS

Future research may concentrate on developing different graph classifiers based on structural patterns. These patterns can be used in combination with popular algorithms solving classification problems e.g. k-Nearest Neighbors (similarity measure based on contrast graphs). Currently, the CCPC is the only classifier that uses contrast graphs. Experiments show that this kind of patterns provide high classification accuracy results for both data representations: decision table and graphs.

CONCLUSION

The most popular methods for solving graph classification problems are based on SVM, k-NN and structural patterns (contrast, common, and frequent graphs). All of these methods are domain independent so they can be used in different areas (where data is represented by a graph).

The recently proposed CCPC algorithm that uses minimal contrast and common graphs outperformed

Table 2.Comparison of classification accuracy for web documents

Document series	Document model, algorithm, parameters	Number of features (dimensions)	Classification accuracy
F	vector, k-NN, cosine graph, k-NN, MCS hybrid, k-NN graph, CCPC	332 30 100 30 / 50 / 100	94.6 *96.8* 95.7 91.4 / **98.9** / **98.9**
J	vector, k-NN, cosine graph, k-NN, MCS hybrid, k-NN graph, CCPC	474 30 / 60 30 / 40 30 / 45	74.6 85.4 / 86.5 87.6 / **94.6** 86.5 / *91.4*
K	vector, k-NN, cosine graph, k-NN, MCS hybrid, k-NN graph, CCPC	1458 40 / 100 / 150 100 / 120 40	77.5 78.2 / 84.6 / *85.7* 86.0 / **86.3** **86.3**

other approaches in terms of accuracy and model complexity for both of analyzed problems: chemical compounds and web documents classification.

REFERENCES

Cook, D. J., & Holder, L. B. (2006). *Mining Graph Data*. Wiley.

De Raedt, L., & Kramer S. (2001). The levelwise version space algorithm and its application to molecular fragment finding. *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, 853-862.

Deshpande, M., Kuramochi, M., & Karypis, G. (2003). Frequent Sub-Structure-Based Approaches for Classifying Chemical Compounds. *Proceedings of 3rd IEEE International Conference on Data Mining (ICDM)*, 25-42.

Diestel, R. (2000). *Graph Theory*. Springer-Verlag.

Dominik, A., & Walczak, Z. (2006). Induction of Decision Rules Using Minimum Set of Descriptors. In L. Rutkowski, R. Tadeusiewicz, L. A. Zadeh, J. Zurada (Ed.), *Proceedings of 8th International Conference on Artificial Intelligence and Soft Computing (ICAISC)*, LNAI 4029, 509-517.

Dominik, A., Walczak Z., & Wojciechowski, J. (2007). Classification of Web Documents Using a Graph-Based Model and Structural Patterns. In J. N. Kok, J. Koronacki, R. López de Mántaras, S. Matwin, D. Mladenic, A. Skowron (Ed.), *Proceedings of 11th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, LNAI 4702, 67-78.

Dominik, A., Walczak Z., & Wojciechowski, J. (2007). Classifying Chemical Compounds Using Contrast and Common Patterns. In B. Beliczynski, A. Dzielinski, M. Iwanowski, B. Ribeiro (Ed.), *Proceedings of 8th International Conference on Adaptive and Natural Computing Algorithms (ICANNGA)*, LNCS 4432 (2), 772-781.

Kotagiri, R., & Bailey, J. (2003). Discovery of Emerging Patterns and their use in Classification. In T. D. Gedeon, L. C. C. Fung (Ed.), *Proceedings of Australian Conference on Artificial Intelligence,* LNCS 2903, 1-12.

Fortin, S. (1996). *The Graph Isomorphism Problem*. Technical report, University of Alberta, Edomonton, Alberta, Canada.

King, R. D., Muggleton, S., Srinivasan, A., Sternberg, M. J. E. (1996). Structure-Activity Relationships Derived by Machine Learning: The use of Atoms and their Bond Connectivities to Predict Mutagenicity by Inductive Logic Programming. *Proceedings of the National Academy of Sciences*, 93, 438-442.

Kozak, K., Kozak, M., & Stapor, K. (2007). Kernels for Chemical Compounds in Biological Screening. In B. Beliczynski, A. Dzielinski, M. Iwanowski, B. Ribeiro (Ed.), *Proceedings of 8th International Conference on Adaptive and Natural Computing Algorithms (ICANNGA)*, LNCS 4432 (2), 327-337.

Kuramochi, M., & Karypis, G. (2001). Frequent Subgraph Discovery. *Proceedings of the 2001 IEEE International Conference on Data Mining (ICDM 2001)*, 313-320.

Markov, A., & Last, M. (2005). Efficient Graph-Based Representation of Web Documents. *Proceedings of the Third International Workshop on Mining Graphs, Trees and Sequences (MGTS 2005)*, 52-62.

Markov, A., Last, M., & Kandel, A. (2006). Model-Based Classification of Web Documents Represented by Graphs. *Proceedings of WebKDD: KDD Workshop on Web Mining and Web Usage Analysis, in conjunction with the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2006)*.

Swamidass, S. J., Chen, J., Bruand, J., Phung, P., Ralaivola, L., & Baldi, P. (2005) Kernels for Small Molecules and the Prediction of Mutagenicity, Toxicity and Anti-cancer Activity. *BioInformatics*, 21(1), 359-368.

Ting, R. M. H., & Bailey, J. (2006). Mining Minimal Contrast Subgraph Patterns. In J. Ghosh, D. Lambert, D. B. Skillicorn, J. Srivastava (Ed.), *Proceedings of 6th International Conference on Data Mining (SIAM)*.

Ullman, J. R. (1976). An Algorithm for Subgraph Isomorphism. *Journal of the ACM*, 23(1), 31-42.

Datasets: Mutag, PTC, NCI. Retrieved October 15, 2007, from http://cdb.ics.uci.edu /CHEMDB/Web/index.htm

Datasets: PDDPdata. Retrieved October 15, 2007, from ftp://ftp.cs.umn.edu /dept/users/boley/PDDPdata/

KEY TERMS

Contrast Graph: Graph that is subgraph isomorphic to at least one graph from particular class of graphs and is not subgraph isomorphic to any graph from any other class of graphs.

Contrast Pattern: Pattern (substructure) that appears in one class of objects and does not appear in other classes.

Common Graph: Graph that is subgraph isomorphic to at least one graph in at least two different classes of graphs.

Common Pattern: Pattern (substructure) that appears in at least two different classes of objects.

Frequent Graph: Graph which support is greater than specified threshold.

Graph Isomorphism: The procedure of testing whether any two graphs are isomorphic (really the same) i.e. finding a mapping from one set of vertices to another set. The complexity of this problem is yet not known to be either P or NP-complete.

Graph Mining: Mining (discovering relevant information) data that is represented as a graph.

Subgraph Isomorphism: The procedure of determining if one smaller graph is included in other larger graph. This problem is proved to be NP-complete.

Subgraph of a Graph G: Graph whose vertex and edge sets are subsets of those of G.

Support of a Graph: The number of graphs to which a particular graph is subgraph isomorphic.

Classifying Two-Class Chinese Texts in Two Steps

Xinghua Fan
Chongqing University of Posts and Telecommunications, China

INTRODUCTION

Text categorization (TC) is a task of assigning one or multiple predefined category labels to natural language texts. To deal with this sophisticated task, a variety of statistical classification methods and machine learning techniques have been exploited intensively (Sebastiani, 2002), including the Naïve Bayesian (NB) classifier (Lewis, 1998), the Vector Space Model (VSM)-based classifier (Salton, 1989), the example-based classifier (Mitchell, 1996), and the Support Vector Machine (Yang & Liu, 1999).

Text filtering is a basic type of text categorization (two-class TC). There are many real-life applications (Fan, 2004), a typical one of which is the ill information filtering, such as erotic information and garbage information filtering on the web, in e-mails and in short messages of mobile phones. It is obvious that this sort of information should be carefully controlled. On the other hand, the filtering performance using the existing methodologies is still not satisfactory in general. The reason lies in that there exist a number of documents with high degree of ambiguity, from the TC point of view, in a document collection, that is, there is a fuzzy area across the border of two classes (for the sake of expression, we call the class consisting of the ill information-related texts, or, the negative samples, the category of TARGET, and, the class consisting of the ill information-not-related texts, or, the positive samples, the category of Non-TARGET). Some documents in one category may have great similarities with some other documents in the other category, for example, a lot of words concerning love story and sex are likely appear in both negative samples and positive samples if the filtering target is erotic information.

BACKGROUND

Fan et al observed a valuable phenomenon, that is, most of the classification errors result from the documents of falling into the fuzzy area between two categories, and presented a two-step TC method based on Naive Bayesian classifier (Fan, 2004; Fan, Sun, Choi & Zhang, 2005; Fan & Sun, 2006), in which the idea is inspired by the fuzzy area between categories. In the first step, the words with parts of speech verb, noun, adjective and adverb are regarded as candidate feature, a Naive Bayesian classifier is used to classify texts and fix the fuzzy area between categories. In the second step, bi-gram of words with parts of speech verb and noun as feature, a Naive Bayesian classifier same as that in the previous step is used to classify documents in the fuzzy area.

The two-step TC method described above has a shortcoming: its classification efficiency is not well. The reason lies in that it needs word segmentation to extract the features, and at currently, the speed of segmenting Chinese words is not high. To overcome the shortcoming, Fan et al presented an improved TC method that uses the bi-gram of character as feature at the first step in the two-step framework (Fan, Wan & Wang, 2006).

Fan presented a high performance prototype system for Chinese text categorization including a general two-step TC framework, in which the two-step TC method described above is regarded as an instance of the general framework, and then presents the experiments that are used to validate the assumption as the foundation of two-step TC method (Fan, 2006). Chen et al. has extended the two-step TC method to multi-class multi-label English (Chen et al., 2007).

MAIN FOCUS

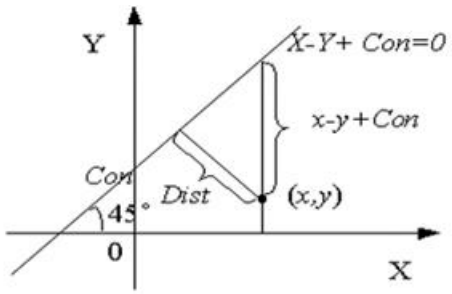
Fix the Fuzzy Area between Categories Using a Naïve Bayesian Classifier

(Fan, 2004; Fan, Sun, Choi & Zhang, 2005; Fan & Sun 2006)

A Naïve Bayesian Classifier is used to fix the fuzzy area in the first step. For a document represented by a binary-valued vector $d=(W_1, W_2, \ldots, W_{|D|})$, the two-class Naïve Bayesian Classifier is given as follows:

$$f(d)=\log\frac{\Pr\{c_1\mid d\}}{\Pr\{c_2\mid d\}} = \log\frac{\Pr\{c_1\}}{\Pr\{c_2\}}+\sum_{k=1}^{|D|}\log\frac{1-p_{k1}}{1-p_{k2}}+\sum_{k=1}^{|D|}W_k\log\frac{p_{k1}}{1-p_{k1}}-\sum_{k=1}^{|D|}W_k\log\frac{p_{k2}}{1-p_{k2}} \quad (1)$$

Figure 1. Distance from point (x,y) to the separate line



where $\Pr\{\cdot\}$ is the probability that event $\{\cdot\}$ occurs, c_i is category i, and $p_{ki}=\Pr\{W_k=1|c_i\}$ (i=1,2). If $f(d)\geq 0$, the document d will be assigned the category label c_1, otherwise, c_2.

Let:

$$Con=\log\frac{\Pr\{c_1\}}{\Pr\{c_2\}}+\sum_{k=1}^{|D|}\log\frac{1-p_{k1}}{1-p_{k2}} \quad (2)$$

$$X=\sum_{k=1}^{|D|}W_k\log\frac{p_{k1}}{1-p_{k1}} \quad (3)$$

$$Y=\sum_{k=1}^{|D|}W_k\log\frac{p_{k2}}{1-p_{k2}} \quad (4)$$

Where *Con* is a constant relevant only to the training set, X and Y are the measures that the document d belongs to categories c_1 and c_2 respectively. (1) is rewritten as:

$$f(d)+X-Y+Con \quad (5)$$

Apparently, $f(d)=0$ is the separate line in a two-dimensional space with X and Y being X-coordinate and Y-coordinate respectively. In this space, a given document d can be viewed as a point (x, y), in which the values of x and y are calculated according to (3) and (4). As shown in Figure1, the distance from the point (x, y) to the separate line will be:

$$Dist=\frac{1}{\sqrt{2}}(x-y+Con) \quad (6)$$

Figure 2. Distribution of the training set in the two-dimensional space

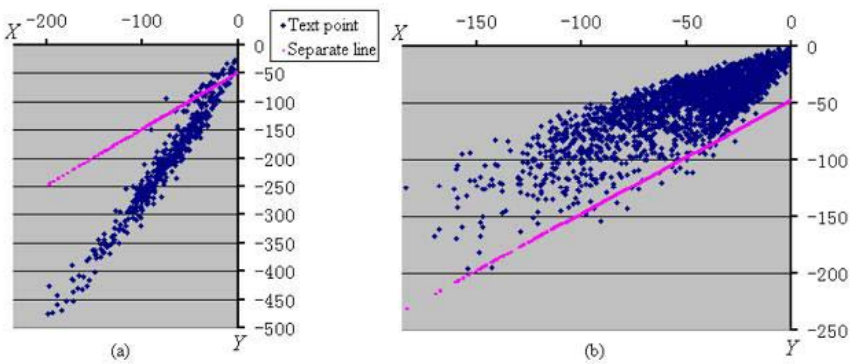


Figure 2 illustrates the distribution of a training set (refer to the next section) regarding *Dist* in the two-dimensional space, with the curve on the left for the negative samples, and the curve on the right for the positive samples. As can be seen in the figure, most of the misclassified documents are near the separate line.

Assumption: the performance of a classifier is relevant to the distance *Dist* in (6), most of the classifying error gathers in an area near the separate line, and the documents falling into this area only are a small portion of the dataset.

Thus, the space can be partitioned into reliable area and unreliable area:

$$\begin{array}{ll} Dist_2 \le Dist \le Dist_1, & \text{Decision for } d \text{ is unreliable} \\ Dist > Dist_1, & \text{Assigning the lable } c_1 \text{ to } d \text{ is reliable} \\ Dist < Dist_2, & \text{Assigning the label } c_2 \text{ to } d \text{ is reliable} \end{array} \quad (7)$$

Where $Dist_1$ and $Dist_2$ are constants determined by experiments, $Dist_1$ is positive real number and $Dist_2$ is negative real number.

Experiments

The experiments include two parts, one is to validate the assumption that is published in Fan (2006), and the other is to valuate the two-step text categorization method that is published in Fan et al. (2005), Fan and Sun (2006), and Fan et al. (2006). In this section, only a part in the latter is given.

Dataset. The dataset used is composed of 12,600 documents with 1,800 negative samples of **TARGET** and 10,800 positive samples of **Non-TARGET**. It is split into 4 parts randomly, with three parts as training set and one part as test set. All experiments are performed in 4-fold cross validation.

Method. Using the naïve as the classifier, reducing features with (8), using five types of features as following: Chinese word, bi-gram of Chinese character, bi-gram of Chinese word, the mixture of Chinese word and bi-gram of Chinese character, and the mixture of Chinese word and bi-gram of Chinese word, doing the seven experiments in table 1, in which every experiment represents a method, the former five methods are one step text categorization and the latter two methods are two step text categorization.

$$MI_1(t_k, c) = \sum_{i=1}^{n} \Pr\{t_k, c_i\} \log \frac{\Pr\{t_k, c_i\}}{\Pr\{t_k\}\Pr\{c_i\}} \quad (8)$$

Table 1. The performance of seven kinds of methods

Exp.	One Step	Two Step	F1 %	Feature Number
1	word	no	91.00	500
2	Bi-gram of Chinese character	no	91.56	800
3	Bi-gram of Chinese word	no	93.65	15000
4	word + Bi-gram of Chinese character	no	96.71	2000
5	word + Bi-gram of Chinese word	no	93.42	800
6	Word	Bi-gram of Chinese word	95.54	500+3000
7	Bi-gram of Chinese character	Bi-gram of Chinese word	97.31	800+8500

Where t_k stands for the kth feature, which may be a Chinese word or a bi-gram of Chinese word, and c_i is the ith-predefined category.

To extract the feature in the second step, CSeg&Tag3.0, a Chinese word segmentation and POS tagging system developed by Tsinghua University, is used to perform the morphological analysis for Chinese texts.

Experimental Results. Only the experimental results for negative samples are considered in evaluation, and the results are showed in Table 1.

Comparing Example 1 and Example 2, it can be seen that Chinese character bi-gram as feature has higher efficiency than Chinese word as feature because it does not need Chinese word segmentation.

Comparing Example 1, Example 3 and Example 5, it can be seen that the bi-gram of Chinese word as feature has better discriminating capability (because the performance of Example 3 and Example 5 is higher than that of Example 1), meanwhile with more serious data sparseness (because the number of features used in Example 3 is more than that used in Example 1 and Example 5)

Comparing Example 5 and Example 3, it can be seen that the mixture of Chinese word and bi-gram of Chinese word as feature is superior to bi-gram of Chinese word as feature because the used feature number in Example 5 is smaller than that in Example 3, and the former has a higher computational efficiency.

Comparing Example 4 and Example 5, it can be seen that the combination of character and word bi-gram is superior to the combination of word and word bi-gram, because the former has better performances.

Comparing Example 6 and Example 7, it can be seen that the method 7 has the best performance and efficiency. Note that Example 7 uses more features, but it does not need Chinese word segmentation in the first step.

Based on experiments and analysis described in above, it shows that bi-gram of Chinese character as feature has better statistic capability than word as feature, so the former has better classification ability in general. But for those documents that have high degree ambiguity between categories, bi-gram of Chinese word as feature has better discriminating capability. So, it obtains high performance if the two types of features are combined to classify Chinese texts in two steps.

Related Works

Combining multiple methodologies or representations has been studied in several areas of information retrieval so far, for example, retrieval effectiveness can be improved by using multiple representations (Rajashekar & Croft, 1995). In the area of text categorization in particular, many methods of combining different classifiers have been developed. For example, Yang et al. (2000) used simple equal weights for normalized score of each classifier output so as to integrate multiple classifiers linearly in the domain of Topic Detection and Tracking; Hull et al. (1996) used linear combination for probabilities or log odds scores of multiple classifier output in the context of document filtering. Larkey and Croft (1996) used weighted linear combination for system ranks and scores of multiple classifier output in the medical document domain; Li and Jain (1998) used voting and classifier selection technique including dynamic classifier selection and adaptive classifier. Lam and Lai (2001) automatically selected a classifier for each category based on the category-specific statistical characteristics. Bennett et al. (2002) used voting, classifier-selection techniques and a hierarchical combination method with reliability indicators.

Comparing with other combination strategy, the two-step method of classifying texts in this chapter has a characteristic: the fuzzy area between categories is fixed directly according to the outputs of the classifier.

FUTURE TRENDS

Our unpublished preliminary experiments show that the two-step TC method is superior to the Support Vector Machine, which is thought as the best TC method in general, not only on the efficiency but also on performance. It is obvious that it needs theory analysis and sufficient experimental validation. At the same time, the possible method of father improving the efficiency is to look for the new feature used in the second step that does not need Chinese word segmentation. I believe that the two-step method can be extended into other language such as English text. A possible solution is to use different type of classifier in the second step. Comparing other classifier combination, this solution will has higher efficiency because only a small part texts in the second step need to process by multiple classifiers.

CONCLUSION

The issue of how to classify Chinese documents with high degree ambiguity is a challenge. A two-step TC approach based on the Naïve Bayesian classifier is developed, which exploits the distributional characteristics of misclassified documents, i.e., most of the misclassified documents are near to the separate line in a constructed two dimensions space. The method has two kinds of implementing editions, one of which exploits the words with parts of speech verb, noun, adjective and adverb as candidate feature at the first step, and the other edition exploits bi-gram of character as candidate feature at the first step. The latter has the best performance and efficiency. The experiments validated the soundness of the two-step method.

It is worth to point out that we believe the two-step method is in principle language independent, though all the experiments are performed on Chinese datasets.

ACKNOWLEDGMENT

The research was supported by the National Natural Science Foundation of China under grant number 60703010, and the Natural Science Foundation of Chongqing province in China under grant number 2006BB2374.

REFERENCES

Bennett, P. N., Dumais, S. T., & Horvitz, E. (2002). Probabilistic Combination of Text Classifiers Using Reliability Indicators: Models and Results. In Proceedings of SIGIR-2002. 11-15.

Jianlin Chen, Xinghua Fan, & Guoyin Wang. (2007). English Texts Categorization in Two-steps. To appear in Guangxi Shifan Daxue Xuebao: Ziran Kexue Ban. 25(4).

Hull, D. A., Pedersen, J. O., & Schutze, H. (1996). Method Combination for Document Filtering. In Proceedings of SIGIR-96. 279–287.

Lam, W., & Lai, K.Y. (2001). A Meta-learning Approach for Text Categorization. In Proceedings of SIGIR-2001. 303-309.

Larkey, L. S., & Croft, W. B. (1996). Combining Classifiers in Text Categorization. In Proceedings of SIGIR-96. 289-297.

Lewis, D. (1998). Naive Bayes at Forty: The Independence Assumption in Information Retrieval. In Proceedings of ECML-98. 4-15.

Li, Y. H., & Jain, A. K. (1998). Classification of Text Documents. The Computer Journal. 41(8), 537-546.

Mitchell, T.M. (1996). Machine Learning. McCraw Hill: New York, NY.

Rajashekar, T. B., & Croft, W. B. (1995). Combining Automatic and Manual Index Representations in Probabilistic Retrieval. Journal of the American society for information science. 6(4), 272-283.

Salton, G. (1989). Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer. Addison-Wesley: Reading, MA.

Sebastiani, F. (2002). Machine Learning in Automated Text Categorization. ACM Computing Surveys. 34(1), 1-47.

Xinghua Fan. (2004). Causality Reasoning and Text Categorization. Postdoctoral Research Report of Tsinghua University, P.R. China.

Xinghua Fan, Maosong Sun, Key-sun Choi, and Qin Zhang. (2005). Classifying Chinese texts in two steps. In Proceedings of IJCNLP-2005, LNAI3651. 302-313.

Xinghua Fan, & Maosong Sun. (2006). A high performance two-class Chinese text categorization method. Chinese Journal of Computers. 29(1), 124-131.

Xinghua Fan, Difei Wan, & Guoyin Wang. (2006). Combining Bi-gram of Character and Word to Classify Two-Class Chinese Texts in Two Steps. In Proceedings of RSCTC-2006, LNAI 4259. 597 – 606.

Xinghua Fan. (2006). A High Performance Prototype System for Chinese Text Categorization. In Proceedings of MICAI 2006, LNAI 4293. 1017 – 1026.

Yang, Y., Ault, T. & Pierce, T. (2000). Combining Multiple Learning Strategies for Effective Cross Validation. In Proceedings of ICML-2000. 1167–1174.

Yang, Y., & Liu, X. (1999). A Re-examination of Text Categorization Methods. In Proceedings of SIGIR-99. 42-49.

KEY TERMS

Information Retrieval (IR): Information retrieval is the art and science of searching for information in documents, searching for documents themselves, searching for metadata which describes documents, or searching within databases, whether relational stand alone databases or hypertext networked databases such as the Internet or intranets, for text, sound, images or data.

High Degree of Ambiguity Text: A document collection, in which there is a fuzzy area across the border of several classes, and some documents in one category may have great similarities with some other documents in the other categories.

Text Categorization (TC): Text categorization is a task of assigning one or multiple predefined category labels to natural language texts.

Text Filtering: It is an information seeking process in which documents are selected from a dynamic text stream to satisfy a relatively stable and specific information need.

Two-Step Classifying Strategy: Two-step classifying strategy exploits two steps to classify text, firstly, a classifier is used to fix the fuzzy area between categories and classify the documents that are not falling into the fuzzy area, and then another classifier that has subtle and powerful features is used to classify the documents that are in the fuzzy area.

Cluster Analysis for Outlier Detection

Frank Klawonn
University of Applied Sciences Braunschweig/Wolfenbuettel, Germany

Frank Rehm
German Aerospace Center, Germany

INTRODUCTION

For many applications in knowledge discovery in databases finding outliers, rare events, is of importance. Outliers are observations, which deviate significantly from the rest of the data, so that it seems they are generated by another process (Hawkins, 1980). Such outlier objects often contain information about an untypical behavior of the system.

However, outliers bias the results of many data mining methods like the mean value, the standard deviation or the positions of the prototypes of *k-means* clustering (Estivill-Castro, 2004; Keller, 2000). Therefore, before further analysis or processing of data is carried out with more sophisticated data mining techniques, identifying outliers is a crucial step. Usually, data objects are considered as outliers, when they occur in a region of extremely low data density.

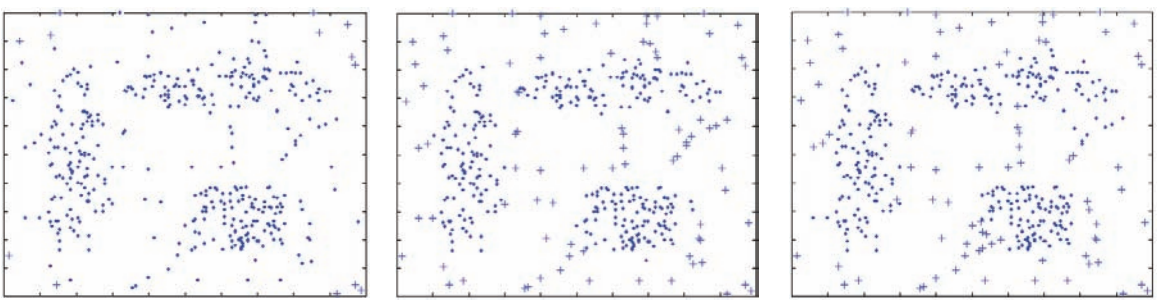
Many clustering techniques like possibilistic clustering (PCM) (Krishnapuram & Keller, 1993; Krishnapuram & Keller, 1996) or noise clustering (NC) (Dave, 1991; Dave & Krishnapuram, 1997) that deal with noisy data and can identify outliers, need good initializations or suffer from lack of adaptability to different cluster sizes (Rehm, Klawonn & Kruse, 2007). Distance-based approaches (Knorr, 1998; Knorr, Ng & Tucakov, 2000) have a global view on the data set. These algorithms can hardly treat data sets containing regions with different data density (Breuning, Kriegel, Ng & Sander, 2000).

In this work we present an approach that combines a fuzzy clustering algorithm (Höppner, Klawonn, Kruse & Runkler, 1999) (or any other prototype-based clustering algorithm) with statistical distribution-based outlier detection.

BACKGROUND

Prototype-based clustering algorithms approximate a feature space by means of an appropriate number of prototype vectors where each prototype vector is located in the center of the group of data *(the cluster)* that belongs to the respective prototype. Clustering usually aims at partitioning a data set into groups or clusters of data where data assigned to the same cluster are similar and data from different clusters are dissimilar. With this partitioning concept in mind, in typical applications of cluster analysis an important aspect is the identification of the number of clusters in a data set. However, when we are interested in identifying outliers, the exact number of clusters is irrelevant (Georgieva & Klawonn, 2006). If one prototype covers two or more data clusters or if two or more prototypes

Figure 1. Outlier detection with different number of prototypes



(a) 1 prototype (b) 3 prototypes (c) 10 prototypes

compete for the same data cluster, this is not important as long as the actual outliers are identified and note assigned to a proper cluster. The number of prototypes used for clustering depends of course on the number of expected clusters but also on the distance measure respectively the shape of the expected clusters. Since this information is usually not available, it is often recommended to use the Euclidean distance measure with rather copious prototypes.

One of the most referred statistical tests for outlier detection is the Grubbs' test (Grubbs, 1969). This test is used to detect outliers in a univariate data set. Grubbs' test detects one outlier at a time. This outlier is removed from the data set and the test is iterated until no outliers are detected.

The detection of outliers as we propose in this work is a modified version of the one proposed in (Santos-Pereira & Pires, 2002) and is composed of two different techniques. In the first step we partition the data set with the fuzzy c-means clustering algorithm so that the feature space is approximated with an adequate number of prototypes. The prototypes will be placed in the center of regions with a high density of feature vectors. Since outliers are far away from the typical data they influence the placing of the prototypes.

After partitioning the data, only the feature vectors belonging to each single cluster are considered for the detection of outliers. For each attribute of the feature vectors of the considered cluster, the mean value and the standard deviation has to be calculated. For the vector with the largest distance[a] to the mean vector, which is assumed to be an outlier, the value of the z-transformation for each of its components is compared to a critical value. If one of these values is higher than the respective critical value, than this vector is declared as an outlier. One can use the Mahalanobis distance as in (Santos-Pereira & Pires, 2002), however since simple clustering techniques like the (fuzzy) c-means algorithm tend to spherical clusters, we apply a modified version of Grubbs' test, not assuming correlated attributes within a cluster.

The critical value is a parameter that must be set for each attribute depending on the specific definition of an outlier. One typical criterion can be the maximum number of outliers with respect to the amount of data (Klawonn, 2004). Eventually, large critical values lead to smaller numbers of outliers and small critical values lead to very compact clusters. Note that the critical value is set for each attribute separately. This leads to an axes-parallel view of the data, which in cases of axes-parallel clusters leads to a better outlier detection than the (hyper)-spherical view on the data.

If an outlier was found, the feature vector has to be removed from the data set. With the new data set, the mean value and the standard deviation have to be calculated again for each attribute. With the vector that has the largest distance to the new center vector, the outlier test will be repeated by checking the critical values. This procedure will be repeated until no outlier will be found anymore. The other clusters are treated in the same way.

Results

Figure 1 shows the results of the proposed algorithm on an illustrative example. The crosses in this figure are feature vectors, which are recognized as outliers. As expected, only few points are declared as outliers, when approximating the feature space with only one prototype. The prototype will be placed in the center

Table 1. Estimated flight duration before and after outlier treatment

Cluster	mean flight duration (s) (before outlier test)	RMSE	mean flight duration (s) (after outlier test)	RMSE
1	2021.18	266.17	2021.18	266.17
2	2497.13	407.90	2465.71	358.68
3	2136.85	268.93	2136.85	268.93
4	2303.41	409.35	2303.41	409.35
5	2186.22	292.04	2186.22	292.04
6	1872.23	180.45	1872.23	180.45
7	2033.31	395.33	2033.31	395.33
8	1879.28	187.12	1879.28	187.12
9	1839.65	90.95	1839.65	90.95
10	2566.15	517.01	2523.28	492.60

of all feature vectors. Hence, only points on the edges are defined as outliers. Comparing the solutions with three and ten prototypes one can determine that both solutions are almost identical. Even in the border regions, were two prototypes compete for some data points, the algorithm rarely identifies these points as outliers, which intuitively are not.

We applied the above method on a weather data set describing the weather situation at a major European airport. Partitioning the weather data is done using fuzzy c-means with 10 prototypes. Since the weather data set is high-dimensional in the feature space we prescind here from showing a visualization of the clustering results. Though, table 1 shows some numerical results for the flight duration before the outlier treatment and afterwards. The proposed outlier procedure removes a total of four outliers in two clusters. The according

clusters are highlighted in the table by means of light grey background. Indeed, both clusters benefit from removing the outliers insofar that the estimation of the flight duration, using a simple measure like the mean, can be improved to a considerable extent. The lower RMSE for the flight duration estimation in both clusters confirms this.

FUTURE TRENDS

The above examples show, that the algorithm can identify outliers in multivariate data in a stable way. With only few parameters the solution can be adapted to different requirements concerning the specific definition of an outlier. With the choice of the number of prototypes, it is possible to influence the result in that way that with lots of prototypes even smaller data groups can be found. To avoid an overfitting to the data it makes sense in certain cases, to eliminate very small clusters. However, finding out the proper number of prototypes should be of interest of further investigations.

In case of using a fuzzy clustering algorithm like FCM (Bezdek, 1981) to partition the data, it is possible to assign a feature vector to different prototype vectors. In that way one can consolidate that a certain feature vector is an outlier or not, if the algorithm decides for each single cluster that the corresponding feature vector is an outlier.

FCM provides membership degrees for each feature vector to every cluster. One approach could be, to assign a feature vector to the corresponding clusters with the two highest membership degrees. The feature vector is considered as an outlier if the algorithm makes the same decision in both clusters. In cases where the algorithm gives no definite answers, the feature vector can be labeled and processed by further analysis.

CONCLUSION

In this work, we have described a method to detect outliers in multivariate data. Since information about the number and shape of clusters is often not known in advance, it is necessary to have a method which is relatively robust with respect to these parameters. To obtain a stable algorithm, we combined approved clustering techniques like the FCM or k-means with a statistical method to detect outliers. Since the complexity of the presented algorithm is linear in the number of points, it can be applied to large data sets.

REFERENCES

Bezdek, J.C. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum Press.

Breunig, M., Kriegel, H.-P., Ng, R.T., Sander, J. (2000). LOF: identifying density-based local outliers. *Proceedings ACM SIGMOD International Conference on Management of Data (SIGMOD 2000),* 93-104.

Dave, R.N. (1991). Characterization and detection of noise in clustering. *Pattern Recognition Letters, 12*, 657-664.

Dave, R.N., Krishnapuram, R. (1997). Robust Clustering Methods: A Unified View. *IEEE Transactions Fuzzy Systems, 5*(2), 270-293.

Estivill-Castro, V., Yang, J. (2004). Fast and robust general purpose clustering algorithms. *Data Mining and Knowledge Discovery, 8, 127–150*, Netherlands: Kluwer Academic Publishers.

Georgieva, O., Klawonn, F. (2006). Cluster analysis via the dynamic data assigning assessment algorithm. *Information Technologies and Control, 2*, 14-21.

Grubbs, F. (1969). Procedures for detecting outlying observations in samples. *Technometrics, 11*(1), 1-21.

Hawkins, D. (1980). *Identification of Outliers.* London: Chapman and Hall.

Höppner, F., Klawonn, F., Kruse, R., Runkler, T. (1999). *Fuzzy Cluster Analysis*. Chichester: John Wiley & Sons.

Keller, A. (2000). Fuzzy Clustering with outliers. In T. Whalen (editor) *19th International Conference of North American Fuzzy Information Processing Society (NAFIPS)*, Atlanta, Georgia: PeachFuzzy.

Klawonn, F. (2004). Noise clustering with a fixed fraction of noise. In: A. Lotfi & J.M. Garibaldi *Applications and Science in Soft Computing,* Berlin, Germany: Springer.

Knorr, E.M., Ng, R.T. (1998). Algorithms for mining distance-based outliers in large datasets. *Proceedings of the 24th International Conference on Very Large Data Bases,* 392-403.

Knorr, E.M., Ng, R.T., Tucakov, V. (2000). Distance-based outliers: algorithms and applications. *VLDB Journal: Very Large Data Bases, 8(3–4)*, 237–253.

Krishnapuram, R., Keller, J. M. (1993). A Possibilistic Approach to Clustering. *IEEE Transactions on Fuzzy Systems, 1*(2), 98-110.

Krishnapuram, R., Keller, J.M. (1996). The possibilistic c-means algorithm: insights and recommendations. *IEEE Transactions on Fuzzy Systems, 4(3)*, 385–393.

Rehm, F., Klawonn, F., Kruse, R. (2007). A novel approach to noise clustering for outlier detection. *Soft Computing, 11*(5), 489-494.

Santos-Pereira, C. M., Pires, A. M. (2002). Detection of outliers in multivariate data: A method based on clustering and robust estimators. In W. Härdle & B. Rönz (editors) *Proceedings in Computational Statistics: 15th Symposium held in Berlin,* Heidelberg, Germany: Physica-Verlag, 291-296.

KEY TERMS

Cluster Analysis: Partition a given data set into clusters where data assigned to the same cluster should be similar, whereas data from different clusters should be dissimilar.

Cluster Prototypes (Centers): Clusters in objective function-based clustering are represented by prototypes that define how the distance of a data object to the corresponding cluster is computed. In the simplest case a single vector represents the cluster, and the distance to the cluster is the Euclidean distance between cluster center and data object.

Fuzzy Clustering: Cluster analysis where a data object can have membership degrees to different clusters. Usually it is assumed that the membership degrees of a data object to all clusters sum up to one, so that a membership degree can also be interpreted as the probability the data object belongs to the corresponding cluster.

Noise Clustering: An additional noise cluster is induced in objective function-based clustering to collect the noise data or outliers. All data objects are assumed to have a fixed (large) distance to the noise cluster, so that only data far away from all other clusters will be assigned to the cluster.

Outliers: are defined as observations in a sample, so far separated in value from the remainder as to suggest that they are generated by another process, or the result of an error in measurement.

Overfitting: is the phenomenon that a learning algorithm adapts so well to a training set, that the random disturbances in the training set are included in the model as being meaningful. Consequently, as these disturbances do not reflect the underlying distribution, the performance on the test set, with its own, but definitively other disturbances, will suffer from techniques that tend to fit to well to the training set.

Possibilistic Clustering: The Possibilistic C-Means (PCM) family of clustering algorithms is designed to alleviate the noise problem by relaxing the constraint on memberships used in probabilistic fuzzy clustering.

Z-Transformation: With the z-transformation one can transform the values of any variable into values of the standard normal distribution.

ENDNOTES

1 To determine the vector with the largest distance to the center vector different distance measures can

be used. For elliptical non axes-parallel clusters, the Mahalanobis distance leads to good results. If no information about the shape of the clusters is available, the Euclidean distance is commonly used.

Cluster Analysis in Fitting Mixtures of Curves

Tom Burr
Los Alamos National Laboratory, USA

INTRODUCTION

One data mining activity is cluster analysis, which consists of segregating study units into relatively homogeneous groups. There are several types of cluster analysis; one type deserving special attention is clustering that arises due to a mixture of curves. A mixture distribution is a combination of two or more distributions. For example, a bimodal distribution could be a mix with 30% of the values generated from one unimodal distribution and 70% of the values generated from a second unimodal distribution.

The special type of mixture we consider here is a mixture of curves in a two-dimensional scatter plot. Imagine a collection of hundreds or thousands of scatter plots, each containing a few hundred points including background noise but also containing from zero to four or five bands of points, each having a curved shape. In one application (Burr et al. 2001), each curved band of points was a potential thunderstorm event (see Figure 1) as observed from a distant satellite and the goal was to cluster the points into groups associated with thunderstorm events. Each curve has its own shape, length, and location, with varying degrees of curve overlap, point density, and noise magnitude. The scatter plots of points from curves having small noise resemble a smooth curve with very little vertical variation from the curve, but there can be a wide range in noise magnitude so that some events have large vertical variation from the center of the band. In this context, each curve is a cluster and the challenge is to use only the observations to estimate how many curves comprise the mixture, plus their shapes and locations. To achieve that goal, the human eye could train a classifier by providing cluster labels to all points in example scatter plots. Each point would either belong to a curved-region or to a catch-all noise category and a specialized cluster analysis would be used to develop an approach for labeling (clustering) the points generated according to the same mechanism in future scatter plots.

BACKGROUND

Two key features that distinguish various types of clustering approaches are the assumed mechanism for how the data is generated and the dimension of the data. The data-generation mechanism includes deterministic and stochastic components and often involves deterministic mean shifts between clusters in high dimensions. But there are other settings for cluster analysis. The particular one discussed here involves identifying thunderstorm events from satellite data as described in the Introduction. From the four examples in Figure 1, note that the data can be described as a mixture of curves where any notion of a cluster mean would be quite different from that in more typical clustering applications. Furthermore, although finding clusters in a two-dimensional scatter plot seems less challenging than in higher-dimensions (the trained human eye is likely to perform as well as any machine-automated method, although the eye would be slower), complications include: overlapping clusters, varying noise magnitude, varying feature and noise and density, varying feature shape, locations, and length, and varying types of noise (scene-wide and event-specific). Any one of these complications would justify treating the fitting of curve mixtures as an important special case of cluster analysis.

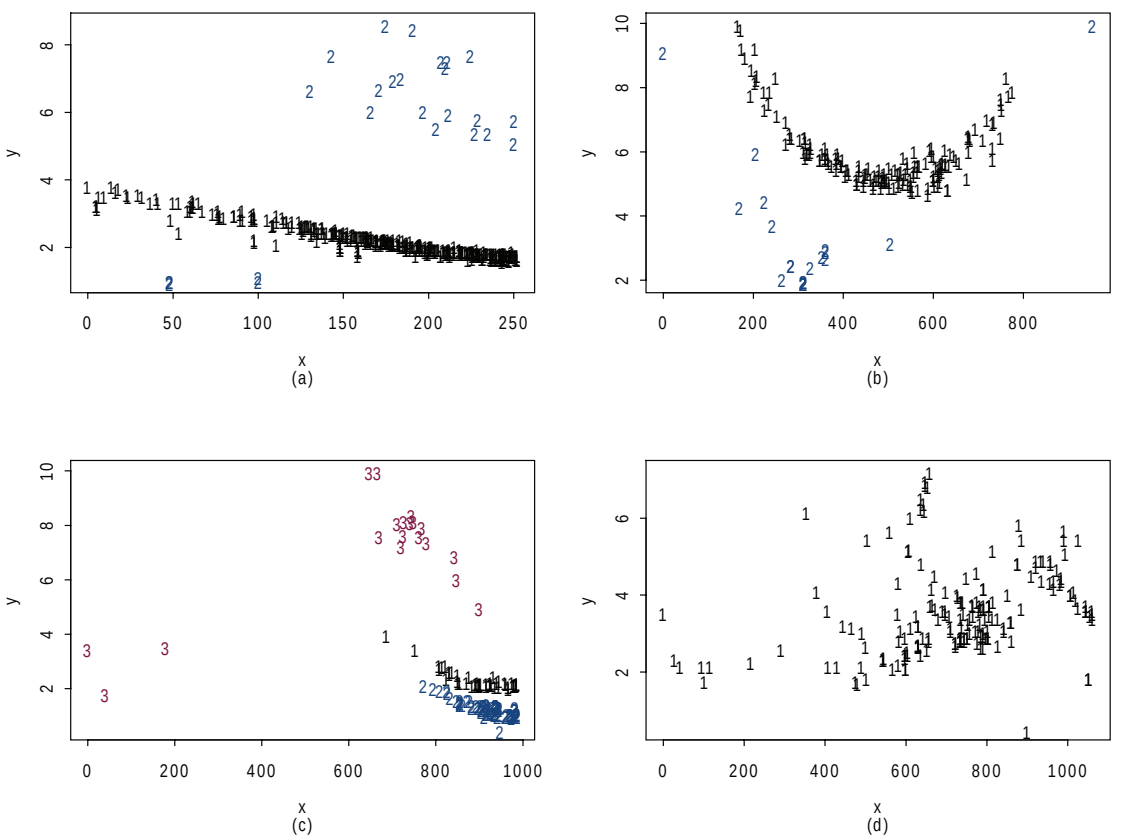
Although as in pattern recognition, the methods discussed below require training scatter plots with points labeled according to their cluster memberships, we regard this as cluster analysis rather than pattern recognition because all scatter plots have from zero to four or five clusters whose shape, length, location, and extent of overlap with other clusters varies among scatter plots. The training data can be used to both train clustering methods, and then judge their quality. Fitting mixtures of curves is an important special case that has received relatively little attention to date. Fitting mixtures of probability distributions dates to Titterington et al. (1985), and several model-based clustering schemes have been developed (Banfield and

Raftery, 1993, Bensmail et al., 1997 and Dasgupta and Raftery, 1998) along with associated theory (Leroux, 1992). However, these models assume that the mixture is a mixture of probability distributions (often Gaussian, which can be long and thin, ellipsoidal, or more circular) rather than curves. More recently, methods for mixtures of curves have been introduced, including a mixture of principal curves model (Stanford and Raftery, 2000), a mixture of regressions model (Turner, 2000; Gaffney and Smyth 2003, and Hurn, Justel, and Robert, 2003), and mixtures of local regression models (smooth curves obtained using splines or nonparametric kernel smoothers for example)

MAIN THRUST OF THE CHAPTER

We describe four methods have been proposed for fitting mixtures of curves. In method 1 (Burr et al., 2001), density estimation is used to reject the background noise points such as those labeled as 2 in Figure 1a. For example, each point has a distance to its *k*th nearest neighbor, which can be used as a local density estimate (Silverman, 1986) to reject noise points. Next, use a distance measure that favors long thin clusters (for example, let the distance between clusters be the minimum distance between a point in the first cluster and a point in the second cluster) together with standard hierarchical clustering to identify at least the central portion of each cluster. Alternatively, model-based clustering favoring long, thin Gaussian shapes (Banfield and Raftery, 1993) or the "fitting straight lines" method in Campbell et al. (1997) are effective for finding the central portion of each cluster. A curve fitted to this central portion can be extrapolated and then used to accept other points as members of the cluster. Because hierarchical clustering cannot accommodate overlapping clusters, this method assumes that the central

Figure 1. Four mixture examples containing (a) one, (b) one, (c) two, and (d) zero thunderstorm events plus background noise. The label "1" is for the first thunderstorm in the scene, "2" for the second, etc., and the highest integer label is reserved for the catch-all "noise" class. Therefore, in (d), because the highest integer is 1, there is no thunderstorm present (the "mixture" is all noise)



portions of each cluster are non-overlapping. Points away from the central portion from one cluster that lie close to the curve fitted to the central portion of the cluster can overlap with points from another cluster. The noise points are initially identified as those having low local density (away from the central portion of any cluster), but during the extrapolation, can be judged to be a cluster member if they lie near the extrapolated curve. To increase robustness, method 1 can be applied twice, each time using slightly different inputs (such as the decision threshold for the initial noise rejection and the criteria for accepting points into a cluster than are close to the extrapolated region of the cluster's curve). Then, only clusters that are identified both times are accepted.

Method 2 uses the minimized integrated squared error (ISE, or L_2 distance) (Scott, 2002, and Scott & Szewczyk, 2002) and appears to be a good approach for fitting mixture models, including mixtures of regression models as is our focus here. Qualitatively, the minimum L_2 distance method tries to find the largest portion of the data that matches the model. In our context, at each stage, the model is all the points belonging to a single curve plus everything else. Therefore, we first seek cluster 1 having the most points, regard the remaining points as noise, remove the cluster and then repeat the procedure in search of feature 2, and so on until a stop criterion is reached. It should also be possible to estimate the number of components in the mixture in the first evaluation of the data but we that approach has not yet been attempted. Scott (2002) has shown that in the parametric setting with model $f(x|\theta)$, we estimate θ using $\hat{\theta} = \arg\min_\theta \int [f(x|\theta) - f(x|\theta_0)]^2 dx$ where the true parameter θ_O is unknown. It follows that a reasonable estimator minimizing the parametric ISE criterion is $\hat{\theta}_{L_2E} = \arg\min_\theta [\int f(x|\theta)^2 dx - \frac{2}{n}\sum_{i=1}^{n} f(x_i|\theta)]$. This assumes that the correct parametric family is used; the concept can be extended to include the case in which the assumed parametric form is incorrect in order to achieve robustness.

Method 3 (Principal curve clustering with noise) was developed by Stanford and Raftery (2000) to locate principal curves in noisy spatial point process data. Principal curves were introduced by Hastie and Stuetzle (1989). A principal curve is a smooth curvilinear summary of *p*-dimensional data. It is a nonlinear generalization of the first principal component line that uses a local averaging method. Stanford and Raftery (2000) developed an algorithm that first uses hierarchical principal curve clustering (HPCC, which is a hierarchical and agglomerative clustering method) and next uses iterative relocation (reassign points to new clusters) based on the classification estimation-maximization (CEM) algorithm. A probability model included the principal curve probability model for the feature clusters and a homogeneous Poisson process model for the noise cluster. More specifically, let X denote the set of observations, $x_1, x_2, \ldots, x_n$ and C be a partition considering of clusters $C_0, C_1, \ldots, C_K$, where the cluster C_j contains n_j points. The noise cluster is C_0 and assume feature points are distributed uniformly along the true underlying feature so there projections onto the feature's principal curve a randomly drawn from a uniform $U(0,\nu_j)$ distribution, where ν_j is the length of the *j*th curve. An approximation to the probability for 0, 1, …, 5 clusters is available from the Bayesian Information Criterion (BIC), which is defined as BIC= $2\log(L(X|\theta) - M\log(n)$, where L is the likelihood of the data X, and M is the number of fitted parameters, so $M = K(DF + 2) + K + 1$. For each of K features we fit 2 parameters (ν_j and σ_j defined below) and a curve having DF degrees of freedom; there are K mixing proportions (π_j defined below) and the estimate of scene area is used to estimate the noise density. The likelihood L satisfies $L(X|\theta) = L(X|\theta) = \prod_{i=1}^{n} L(x_i|\theta)$ where $L(x_i|\theta) = \sum_{j=0}^{K} \pi_j L(x_i|\theta, x_i \in C_j)$ is the mixture likelihood (π_j is the probability that point *i* belongs to feature *j*) $L(x_i|\theta, x_i \in C_j) = (1/\nu_j)(1/\sqrt{2\pi}\sigma_j)\exp(\frac{-\|x_i - f(\lambda_{ij})\|^2}{2\sigma_j^2})$ and for the feature clusters ($\|x_i - f(\lambda_{ij})\|$ is the Euclidean distance from x_i to its projection point $f(\lambda_{ij})$ on curve j) and $L(x_i|\theta, x_i \in C_j) = 1/Area$ for the noise cluster. Space will not permit a complete description of the HPCC-CEM method, but briefly, the HPCC steps are: (1) Make an initial estimate of noise points and remove then; (2) Form an initial clustering with at least seven points in each cluster; (3) Fit a principal curve to each cluster; (4) Calculate a clustering criterion $V = V_{About} + \alpha V_{Along}$, where V_{About} measures the orthogonal distances to the curve ("residual error sum of squares") and V_{Along} measures the variance in arc length distances between projection points on the curve. Minimizing V (the sum is over all clusters) will lead to clusters with regularly spaced points along the curve, and tightly grouped around it. Large values of α will cause the method to avoid clusters with gaps and small values of α favor thinner clusters. Clustering

(merging clusters) continues until V stops decreasing. The flexibility provided by such a clustering criterion (avoid or allow gaps and avoid or favor thin clusters) is useful and Method 3 is currently the only published method to include it.

Method 4 (Turner, 2000) uses the well-known EM (estimation – maximization) algorithm (Dempster, Laird, and Rubin, 1977) to handle a mixture of regression models. Therefore, it is similar to method 3 in that a mixture model is specified, but differs in that the curve is fit using parametric regression rather than principal curves, so

$$L(x_i \mid \theta, x_i \in C_j) = f_{ij} = (1/\sigma_j)\,\phi\left(\frac{y_i - x_i\beta_j}{\sigma_j}\right)$$

where ϕ is the standard Gaussian distribution. Also, Turner's implementation did not introduce a clustering criterion, but it did attempt to estimate the number of clusters as follows. Introduce indicator variable z_i of which component of the mixture generated observation y_i and iteratively maximize

$$Q = \sum_{i=1}^{n}\sum_{k=1}^{K}\gamma_{ik}\ln(f_{ik})$$

with respect to θ where θ is the complete parameter set π_j for each class. The γ_{ik} satisfy

$$\gamma_{ik} = \pi_k f_{ik} / \sum_{i=1}^{K}\pi_k f_{ik}.$$

Then Q is maximized with respect to t by weighted regression of y_i on $x_1, \ldots, x_n$ with weights γ_{ik} and each s_k^2 is given by

$$\sigma_k^2 = \sum_{i=1}^{n}\lambda_{ik}(y_i - x_i\beta_k)^2 / \sum_{i=1}^{n}\gamma_{ik}.$$

In practice the components of θ come from the value of θ in the previous iteration and

$$\pi_k = (1/n)\sum_{i=1}^{n}\gamma_{ik}.$$

A difficulty in choosing the number of components in the mixture, each with unknown mixing probability, is that the likelihood ratio statistic has an unknown distribution. Therefore Turner (2000) implemented a bootstrap strategy to choose between 1 and 2 components, between 2 and 3, etc. The strategy to choose between K and $K + 1$ components is: (a) calculate the log-likelihood ratio statistic Q for a model having K and for a model having $K + 1$ components; (b) simulate data from the fitted K -component model; (c) fit the K and $K + 1$ component Q*; (d) compute the p-value for Q as

$$p = 1/n\sum_{i=1}^{n} I(Q \geq Q^*),$$

where the indicator $I(Q \geq Q^*) = 1$ if $Q \geq Q^*$ and 0 otherwise.

To summarize, method 1 avoids the mathematics of mixture fitting and seeks high-density regions to use as a basis for extrapolation. It also checks the robustness of its solution by repeating the fitting using different input parameters and comparing results. Method 2 uses a likelihood for one cluster and "everything else," then removes the highest-density cluster, and repeats the procedure. Methods 3 and 4 both include formal mixture fitting (known to be difficult), with method 3 assuming the curve is well modeled as a principal curve and method 4 assuming the curve is well fit using parametric regression. Only method 3 allows the user to favor or penalize clusters with gaps. All methods can be tuned to accept only thin (small residual variance) clusters.

FUTURE TRENDS

Although we focus here on curves in the two dimensions, clearly there are analogous features in higher dimensions, such as principal curves in higher dimensions. Fitting mixtures in two dimensions is fairly straightforward, but performance is rarely as good as desired. More choices for probability distributions are needed; for example, the noise in Burr et al. (2001) was a non-homogeneous Poisson process, which might be better handled with adaptively chosen regions of the scene. Also, regarding noise removal in the context of mixtures of curves, Maitra and Ramler (2006) introduce some options to identify and remove "scatter" as a preliminary step to traditional cluster analysis. Also, very little has been published regarding the "measure of difficulty" of curve-mixture problem, although Hurn, Justel, and Robert (2003) provided a numerical Bayesian approach for mixtures involving "switching regression." The term "switching regression" describes the situation in which the likelihood is nearly unchanged when some

of the cluster labels are switched. For example, imagine that the clusters 1 and 2 were slightly closer in Figure 1c, and then consider changes to the likelihood if we switched some 2 and 1 labels. Clearly if curves are not well separated, we should not expect high clustering and classification success. In the case where groups are separated by mean shifts, it is clear how to define group separation. In the case where groups are curves that overlap to varying degrees, one could envision a few options for defining group separation. For example, the minimum distance between a point in one cluster and a point in another cluster could define the cluster separation and therefore also the measure of difficulty. Depending on how we define the optimum, is possible that performance can approach the theoretical optimum. We define performance by first determining whether a cluster was found, and then reporting the false positive and false negative rates for each found cluster.

Another area of valuable research would be to accommodate mixtures of local regression models (such as smooth curves obtained using splines or nonparametric kernel smoothers, Green and Silverman, 1996). Because local curve smoothers such as splines are extremely successful in the case where the mixture is known to contain only one curve, it is likely that the flexibility provided by local regression (parametric or nonparametric) would be desirable in the broader context of fitting a mixture of curves.

Existing software is fairly accessible for the methods described, assuming users have access to a statistical programming language such as S-PLUS (2003) or R. Executable code to accommodate a user-specified input format would be a welcome future contribution. A relatively recent R package (Boulerics, 2006), to fit "general nonlinear mixtures of curves" is freely available in R and includes several related functions.

CONCLUSION

Fitting mixtures of curves with noise in two dimensions is a specialized type of cluster analysis. It is a challenging cluster analysis task. Four options have been briefly described. Example performance results for these methods for one application are available (Burr et al., 2001). Results for methods 2-4 are also available in their respective references (on different data sets), and results for a numerical Bayesian approach using a mixture of regressions with attention to the case having nearly overlapping clusters has also been published (Hurn, Justel, and Robert, 2003).

REFERENCES

Banfield, J., & Raftery, A. (1993). Model-based gaussian and non-gaussian clustering. *Biometrics*, *49*, 803-821.

Bensmail, H., Celeux, G., Raftery, A., & Robert, C. (1997). Inference in model-based cluster analysis. *Statistics and Computing, 7*, 1-10.

Boulerics, B., (2006). The MOC Package (mixture of curves), available as an R statistical programming package at http://www.r-project.org

Burr, T., Jacobson, A., & Mielke, A. (2001). Identifying storms in noisy radio frequency data via satellite: an application of density Estimation and cluster analysis, Los Alamos National Laboratory internal report LA-UR-01-4874. In *Proceedings of US Army Conference on Applied Statistics* (CD-ROM), Santa Fe, NM.

Campbell, J., Fraley, C., Murtagh, F., & Raftery, A. (1997). Linear flaw detection in woven textiles using model-based clustering. *Pattern Recognition Letters 18*, 1539-1548.

Dasgupta, A., & Raftery, A. (1998). Detecting features in spatial point processes with clutter via model-based clustering. *Journal of American Statistical Association, 93*(441), 294-302.

Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood for incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39, 1-38.

Gaffney, S., & Smyth, P. (2003). Curve clustering with random effects regression mixtures. In *Proc. 9th Inter. Conf, on AI and Statistics*, Florida.

Green, P., & Silverman, B. (1994). *Nonparametric Regression and Generalized Linear Models*. Chapman and Hall: London.

Hastie, T, & Stuetzle, W. (1989). Principal curves. *Journal of American Statistical Association, 84,* 502-516.

Hurn, M., Justel, A., & Robert, C. (2003). Estimating mixtures of regressions. *Journal of Computational and Graphical Statistics, 12*, 55-74.

Leroux, B. (1992). Consistent estimation of a mixing distribution. *The Annals of Statistics, 20,* 1350-1360.

Maitra, R., & Ramler, I. P. (2006). Clustering in the Presence of Scatter, submitted to *Biometrics*, and presented at 2007 *Spring Research Conference on Statistics in Industry and Technology* available on CD-ROM.

Scott, D. (2002). Parametric statistical modeling by minimum integrated square error. *Technometrics 43*(3), 274-285.

Scott, D., & Szewczyk, W. (2002). From kernels to mixtures. *Technometrics, 43*(3), 323-335.

Silverman, B. (1986). *Density Estimation for Statistics and Data Analysis*. Chapman and Hall: London.

S-PLUS Statistical Programming Language (2003). Insightful Corp, Seattle Washington.

Stanford, D., & Raftery, A. (2000). Finding curvilinear features in spatial point patterns: principal curve clustering with noise, *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22*(6), 601-609.

Tiggerington, D., Smith, A., & Kakov, U. (1985). *Statistical Analysis of Finite Mixture Distributions.* Wiley: New York.

Turner, T. (2000). Estimating the propagation rate of a viral infection of potato plants via mixtures of regressions. *Applied Statistics, 49*(3), 371-384.

KEY TERMS

Bayesian Information Criterion: An approximation to the Bayes Factor which can be used to estimate the Bayesian posterior probability of a specified model.

Bootstrap: A resampling scheme in which surrogate data is generated by resampling the original data or sampling from a model that was fit to the original data.

Probability Density Function Estimate: An estimate of the probability density function. One example is the histogram for densities that depend on one variable (or multivariate histograms for multivariate densities). However, the histogram has known deficiencies involving the arbitrary choice of bin width and locations. Therefore the preferred density function estimator is a smoother estimator that uses local weighted sums with weights determined by a smoothing parameter is free from bin width and location artifacts.

Estimation-Maximization Algorithm: An algorithm for computing maximum likelihood estimates from incomplete data. In the case of fitting mixtures, the group labels are the missing data.

Mixture of Distributions: A combination of two or more distributions in which observations are generated from distribution *i* with probability π_i and $\Sigma\pi_i = 1$.

Principal Curve: A smooth, curvilinear summary of p-dimensional data. It is a nonlinear generalization of the first principal component line that uses a local average of p-dimensional data.

Probability Density Function: A function that can be summed (for discrete-valued random variables) or integrated (for interval-valued random variables) to give the probability of observing values in a specified set.

Cluster Analysis with General Latent Class Model

Dingxi Qiu
University of Miami, USA

Edward C. Malthouse
Northwestern University, USA

INTRODUCTION

Cluster analysis is a set of statistical models and algorithms that attempt to find "natural groupings" of sampling units (e.g., customers, survey respondents, plant or animal species) based on measurements. The observable measurements are sometimes called *manifest* variables and cluster membership is called a *latent* variable. It is assumed that each sampling unit comes from one of K clusters or classes, but the cluster identifier cannot be observed directly and can only be inferred from the manifest variables. See Bartholomew and Knott (1999) and Everitt, Landau and Leese (2001) for a broader survey of existing methods for cluster analysis.

Many applications in science, engineering, social science, and industry require grouping observations into "types." Identifying typologies is challenging, especially when the responses (manifest variables) are categorical. The classical approach to cluster analysis on those data is to apply the latent class analysis (LCA) methodology, where the manifest variables are assumed to be independent conditional on the cluster identity. For example, Aitkin, Anderson and Hinde (1981) classified 468 teachers into clusters according to their binary responses to 38 teaching style questions. This basic assumption in classical LCA is often violated and seems to have been made out of convenience rather than it being reasonable for a wide range of situations. For example, in the teaching styles study two questions are "Do you usually allow your pupils to move around the classroom?" and "Do you usually allow your pupils to talk to one another?" These questions are mostly likely correlated even within a class.

BACKGROUND

This chapter focuses on the mixture-model approach to clustering. A mixture model represents a distribution composed of a mixture of component distributions, where each component distribution represents a different cluster. Classical LCA is a special case of the mixture model method. We fit probability models to each cluster (assuming a certain fixed number of clusters) by taking into account correlations among the manifest variables. Since the true cluster memberships of the subjects are unknown, an iterative estimation procedure applicable to missing data is often required.

The classical LCA approach is attractive because of the simplicity of parameter estimation procedures. We can, however, exploit the correlation information between manifest variables to achieve improved clustering. Magidson and Vermunt (2001) proposed the latent class factor model where multiple latent variables are used to explain associations between manifest variables (Hagenaars, 1988; Magidson & Vermunt, 2001; Hagenaars & McCutcheon, 2007). We will, instead, focus on generalizing the component distribution in the mixture model method.

MAIN FOCUS

Assume a random sample of n observations, where each comes from one of K unobserved classes. Random variable $Y \in \{1, \ldots, K\}$ is the latent variable, specifying the value of class membership. Let $P(Y=k) = \eta_k$ specify the *prior distribution* of class membership, where

$$\sum_{k=1}^{K} \eta_k = 1.$$

For each observation $i = 1, \ldots, n$, the researcher observes p manifest variables $\boldsymbol{X}_i = (X_{i1}, \ldots, X_{ip})'$. Given that an observation comes from class k (i.e., $Y = k$), the *class-conditional distribution* of $\boldsymbol{X}$, denoted as $f_k(\boldsymbol{x};\theta_{\boldsymbol{k}})$, is generally assumed to come from common distribution families. For example, classical LCA assumes that the components of $\boldsymbol{X}$ are each multinomial and independent of each other for objects within the same class. Suppose each manifest variable takes only 2 values, hereafter labeled generically "yes" and "no", then $P(X_j = x_j \mid Y)$ is a Bernoulli trial. Let π_{jk} be the probability that someone in class k has a "yes" value to manifest variable X_j. Then the class-conditional distribution, under the assumption of class-conditional independence, is

$$f_k(\boldsymbol{x};\theta_{\boldsymbol{k}}) = \prod_{j=1}^{p} P(x_j \mid Y=k) = \prod_{j=1}^{p} \pi_{jk}^{x_j} (1-\pi_{jk})^{1-x_j}. \quad (1)$$

This assumption greatly reduces the number of parameters that must be estimated. However, in many cases, more flexible distributions should be developed to allow for improved clustering.

Component Distributions

In general, $f_k(\boldsymbol{x};\theta_{\boldsymbol{k}})$ can take any component distribution. However, due to the constraint of identifiableness and the computing requirement for parameter estimation, only two component distributions, to our best knowledge, have been proposed in the literature to address the correlation structure within each cluster. Qu, Tan and Kutner (1996) proposed a random effects model that is a restricted version of the multivariate Probit model. The conditional dependence is modeled by subject-specific random variables. The manifest variables are correlated because of the correlations between the underlying normal random variables in addition to the class membership. The correlation matrix in the component distribution of the random effects model has a restricted structure which makes it less appealing (Tamhane, Qiu & Ankenman, 2006).

Tamhane, Qiu and Ankenman (2006) provide another general-purpose multivariate Bernoulli distribution, called the continuous latent variable (CLV) model, based on subject-specific uniform random variables. This proposed distribution can handle both positive and negative correlations for each component cluster. It is relatively flexible in the sense that the correlation matrix does not have any structural restrictions as in the random effects model. This approach has been applied to two real data sets (Tamhane, Qiu & Ankenman, 2006) and provided easily interpretable results.

Parameter Estimation

The model parameters (η_k and $\boldsymbol{\theta}_k$) are estimated with maximum likelihood. The probability density function of the mixture is

$$f(\boldsymbol{x};\psi) = \sum_{k=1}^{K} \eta_k f_k(\boldsymbol{x};\theta_k),$$

where the vector ψ of unknown parameters consists of the mixture proportions η_k and class-conditional distribution parameters $\theta_{\boldsymbol{k}}$. Under the assumption that $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$ are independent observations, the incomplete log-likelihood function is given by

$$\log L = \sum_{i=1}^{n} \log \sum_{k=1}^{K} \eta_k f_k(\boldsymbol{x}_i, \theta_k),$$

which must be maximized with respect to parameters η_k and $\boldsymbol{\theta}_k$. Due to the summation inside the logarithm, direct maximization is difficult, and the expectation-maximization (EM) algorithm of Dempster, Laird and Rubin (1977) is generally used to obtain the parameter estimators. See Bartholomew and Knott (1999, pp. 137-139) for details. The EM algorithm is convenient to construct if there exist closed-form solutions to the maximum likelihood estimators (MLEs). When closed-form solutions do not exist, the more general optimization procedures, such as quasi-Newton method, will be used. Generally speaking, there are no known ways of finding starting values that guarantee a global optimum, and different starting values will often produce different local maxima. One solution to the starting-value problem is to run the optimization with multiple random starting values and select the one with the largest log-likelihood value. Commercial software package such as Knitro® and LatentGold® solve this type of optimization problem. There are also software packages in the public domain.[a]

Assignment of Units to Clusters

The researcher often seeks to infer the class membership based on the observed data. This inference is made with the *posterior distribution*, which gives the probability of belonging to class *k* given the observed measurements $\boldsymbol{x}$

$$P(Y=k \mid \boldsymbol{x}) = \frac{f_k(\boldsymbol{x};\theta_k)}{\sum_{h=1}^{K} f_h(\boldsymbol{x};\theta_h)\eta_k}. \qquad (2)$$

Objects are usually assigned by the *maximum posterior rule* to the class with the highest posterior probability computed from (2). Assigning sampling units to a single class is called *hard assignment*. In contrast to the hard assignment, *soft assignment* is based on the probabilities. Sampling units are allowed to have partial membership in multiple classes. Probabilities allow the researcher to distinguish between cases that almost certainly come from a particular class and those that could plausibly come from several.

Number of Clusters

Determining of the number of clusters *K* is a special case of the model selection problem. The problem becomes difficult when the clusters overlap with each other, which is typical in cluster analysis of categorical data. Various approaches have been proposed in the statistics literature to determine the "true" number of clusters in a data set. Examples include the silhouette statistic (Kaufman and Rousseeuw 1990), gap statistics (Tibshirani, Walther, and Hastie 2001) and jump method (Sugar and Kames 2003). However, these statistics are mainly designed for continuous data where the distance measure between objects can easily be defined. For the categorical data, the general-purpose approach is to estimate mixture models for various values of *K* and then compare model summaries such as Akaike's Information Criterion (AIC) and Bayesian Information Criterion (BIC), which are defined as

$$\text{AIC} = -2 \log L + 2m \text{ and BIC} = -2 \log L + m \log n,$$

where *m* is the number of parameters. Both criteria penalize the log-likelihood value based on the number of the parameters in the model. However, the AIC criterion satisfies the optimality condition if the underlying "true" model is infinite dimensional (Hoeting, Davis, Merton, and Thompson, 2004), which is not true for the general latent class model. In addition, the best model selected by AIC depends on the sample size (Burnham and Anderson, 1998). BIC, on the other hand, is consistent in the sense that if the true model is among the candidates, the probability of selecting the true model approaches 1 as the sample size increases (Keribin, 2000). In fact, BIC is the only criterion implemented in the commercial software such as Latent Gold®[b].

In applied contexts, the usefulness of a clusters solution should be considered when determining the number of clusters. However, usefulness is often subjective and the researcher must combine domain knowledge with the fit statistics mentioned earlier to decide the optimal number of clusters. The ultimate objective of a cluster analysis is to identify a typology that is "useful" or "actionable" in some way, and the number of discrete clusters must serve that purpose. See Malthouse (2003, pp. 179-180) and Malthouse and Calder (2005) for specific discussion of this question for customer relationship management (CRM) applications.

Selection of Manifest Variables

The researcher can measure a large number of attributes (manifest variables), but not all should be included as $\boldsymbol{X}$ variables in a cluster analysis. As indicated above, the objective of cluster analysis is to identify "useful" groups. The meaning of useful, however, depends on the particular application. Which variables are used depends on how the typology will be used, not on the data set itself. Consider the task of grouping animals such as mice, elephants, sharks, whales, ostriches, goldfish, and cardinals. The variables that should be used in the analysis depend entirely on the application. Whales, elephants, and mice are all mammals based on them having warm blood, hair, nursing their young, etc. Yet if the objective is to separate "large" animals from "small" ones, manifest variables such as weight, height, and length should be used in the clustering, grouping elephants and whales with sharks and ostriches.

Noisy variables tend to mask the cluster structure of the data and empirical variable selection methods have also been proposed. For example, Raftery and Dean (2006) proposed a general purpose variable selection procedure for model-based clustering. Their application to simulated data and real examples show

that removing irrelevant variables improves clustering performance.

Simulation Study

The performances of different clustering methods are often compared via simulated data because with most real observed data sets there is usually no way to identify the true class membership of observations. Simulated data enables researchers to understand how robust the newly proposed methods are even if the data at hand do not follow the assumed distribution. The performance of a clustering method is often evaluated by a measure called *correct classification rate* (*CCR*) (Tamhane, Qiu and Ankenman 2006), which is the proportion of correctly classified observations. The observation cluster identities are known in advance in simulation, and hence the correctly classified objects can be easily counted.

The CCR has lower (LCCR) and upper bounds (UCCR) that are functions of simulated data (Tamhane, Qiu & Ankenman, 2006). Sometimes, it is better to evaluate the normalized CCR, which is called the correct classification score (CCS):

$$CCS = \frac{CCR - LCCR}{UCCR - LCCR}.$$

The CCS falls between 0 and 1, and larger values of CCS are desirable.

FUTURE TRENDS

Cluster analysis with latent class model is a well-plowed field that is still relatively fast moving. The classical approaches have been implemented in specialized[c] commercial and free software. Recent developments have focused on finding a novel way of modeling conditional dependence structure under the framework of the mixture model. Both the random effects model and the CLV model have been proposed to model the dependence structure in the component clusters. The main limitation of these two methods is that they are computationally intensive. For example, a PC with 2.8 GHz clock speed requires nearly six hours to estimate the parameters in a CLV model with 7 responses and 2 clusters. Recent model developments include the binary latent variable model (Al-Osh & Lee, 2001) and the latent variable model for attitude data measured on Likert scales (Javaras and Ripley, 2007). Future research directions in the general latent class model include constructing new parsimonious distributions to handle the conditional dependence structure of the manifest variables and proposing better parameter estimation methods.

The problem of finding the optimal number of clusters is difficult even for continuous data, where the distance measures between objects can be easily defined. The problem becomes even more acute when all manifest variables are categorical. Opportunities certainly exist for researchers to develop new statistic customized for categorical data.

CONCLUSION

The latent class models can be used as a model-based approach to clustering categorical data. It can classify objects into clusters through fitting a mixture model. Objects are assigned to clusters with the maximum posterior rule. The local independence assumption in the classical LCA is too strong in many applications. However, the classical LCA method can be generalized to allow for modeling the correlation structure between manifest variables. Applications to real data sets (Hadgu and Qu, 1998; Tamhane, Qiu and Ankenman, 2006) show a promising future.

REFERENCES

Aitkin, M., Anderson, D., & Hinde, J. (1981). Statistical modelling of data on teaching styles, *Journal of Royal Statistical Society*, Series A, 144, 419-461.

Al-Osh, M. A., & Lee, S. J. (2001). A simple approach for generating correlated binary variates, *Journal of Statistical Computation and Simulation*, 70, 231-235.

Bartholomew, D. J., & Knott M. (1999). *Latent variable models and factor analysis,* 2nd edition, New York: Oxford University Press, Inc.

Burnham, K. P., & Anderson, D. R. (1998). *Model selection and inference*, New York: Springer-Verlag.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the

EM algorithm (with discussion), *Journal of the Royal Statistical Society*, Series B, 39, 1-38.

Everitt, B. S., Landau, S., & Leese, M. (2001). *Cluster analysis*, 4th Ed., New York: John Wiley & Sons, Inc.

Hagenaars, J. A. (1988). Latent structure models with direct effects between indicators: local dependence models, *Sociological Methods and Research*, 16, 379-405.

Hagenaars, J. A., & McCutcheon, A. L. (2007). *Applied latent class analysis*, New York: Cambridge University Press.

Hadgu, A., & Qu, Y. (1998). A biomedical application of latent class models with random effects, *Applied Statistics*, 47, 603-616.

Hoeting, J. A., Davis, R. A., Merton, A. A., & Thompson, S. E. (2004). Model selection for geostatistical models, *Ecological Applications*, 16(1), 87–98.

Javaras, K. N., & Ripley, B. D. (2007). An "unfolding" latent variable model for likert attitude data: Drawing inferences adjusted for response style, *Journal of the American Statistical Association*, 102 (478), 454-463.

Kaufman, L., & Rousseeuw, P. (1990). *Finding groups in data: An introduction to cluster analysis*, New York: Wiley.

Keribin, C. (2000). Consistent estimation of the order of mixture models, *The Indian Journal of Statistics*, Series A, 1, 49-66.

Magidson, J., & Vermunt J. K. (2001). Latent class factor and cluster models, bi-plots, and related graphical displays, *Sociological Methodology*, 31, 223-264

Malthouse, E. C. (2003). Database sub-segmentation, In Iacobucci D. & Calder B. (Ed.), *Kellogg on integrated marketing* (pp.162-188), Wiley.

Malthouse, E. C., & Calder, B. J. (2005). Relationship branding and CRM, in *Kellogg on Branding*, Tybout and Calkins editors, Wiley, 150-168.

Qu, Y., Tan, M., & Kutner, M. H. (1996). Random effects models in latent class analysis for evaluating accuracy of diagnostic tests, *Biometrics*, 52, 797-810.

Raftery, A. E., & Dean, N. (2006). Variable selection for model-based clustering, *Journal of the American Statistical Association*, 101(473), 168-178.

Sugar, C. A., & James, G. M. (2003). Finding the number of clusters in a dataset: An information-theoretic approach, *Journal of the American Statistical Association*, 98, 750-763.

Tamhane, A. C., Qiu, D., & Ankenman, B. A. (2006). *Latent class analysis for clustering multivariate correlated Bernoulli data*, Working paper No. 06-04, Department of IE/MS, Northwestern University, Evanston, Illinois. (Downloadable from http://www.iems.northwestern.edu/content/Papers.asp)

Tibshirani, R., Walther, G., & Hastie, T. (2001). Estimating the number of clusters in a data set via gap statistic, *Journal of the Royal Statistical Society*, Series B, 63, 411-423.

KEY TERMS

Cluster Analysis: The classification of objects with measured attributes into different unobserved groups so that objects within the same group share some common trait.

Hard Assignment: Observations are assigned to clusters according to the maximum posterior rule.

Latent Variable: A variable that describes an unobservable construct and cannot be observed or measured directly. Latent variables are essential elements of latent variable models. A latent variable can be categorical or continuous.

Manifest Variable: A variable that is observable or measurable. A manifest variable can be continuous or categorical. Manifest variables are defined in contrast with the latent variable.

Mixture Model: A mixture model represents a sample distribution by a mixture of component distributions, where each component distribution represents one cluster. It is generally assumed that data come from a finite number of component distributions with fixed but unknown mixing proportions.

Simulation: An imitation of a real-life or hypothetical situation on a computer so that it can be studied to see how the system or a proposed model works.

Soft Assignment: Probabilities of belonging to different clusters are calculated and objects are assigned to clusters according to these probabilities as if objects can be divided into fractions.

ENDNOTES

[a] See http://ourworld.compuserve.com/homepages/jsuebersax/

[b] http://www.statisticalinnovations.com/products/latentgold_v4.html

[c] It is not currently available in general-purpose statistical and data-mining software.

Cluster Validation

Ricardo Vilalta
University of Houston, USA

Tomasz Stepinski
Lunar and Planetary Institute, USA

INTRODUCTION

Spacecrafts orbiting a selected suite of planets and moons of our solar system are continuously sending long sequences of data back to Earth. The availability of such data provides an opportunity to invoke tools from machine learning and pattern recognition to extract patterns that can help to understand geological processes shaping planetary surfaces. Due to the marked interest of the scientific community on this particular planet, we base our current discussion on Mars, where there are presently three spacecrafts in orbit (e.g., NASA's Mars Odyssey Orbiter, Mars Reconnaissance Orbiter, ESA's Mars Express). Despite the abundance of available data describing Martian surface, only a small fraction of the data is being analyzed in detail because current techniques for data analysis of planetary surfaces rely on a simple visual inspection and descriptive characterization of surface landforms (Wilhelms, 1990).

The demand for automated analysis of Mars surface has prompted the use of machine learning and pattern recognition tools to generate geomorphic maps, which are thematic maps of landforms (or topographical expressions). Examples of landforms are craters, valley networks, hills, basins, etc. Machine learning can play a vital role in automating the process of geomorphic mapping. A learning system can be employed to either fully automate the process of discovering meaningful landform classes using *clustering* techniques; or it can be used instead to predict the class of unlabeled landforms (after an expert has manually labeled a representative sample of the landforms) using *classification* techniques. The impact of these techniques on the analysis of Mars topography can be of immense value due to the sheer size of the Martian surface that remains unmapped.

While it is now clear that machine learning can greatly help in automating the detailed analysis of Mars' surface (Stepinski et al., 2007; Stepinski et al., 2006; Bue and Stepinski, 2006; Stepinski and Vilalta, 2005), an interesting problem, however, arises when an automated data analysis has produced a novel classification of a specific site's landforms. The problem lies on the interpretation of this new classification as compared to traditionally derived classifications generated through visual inspection by domain experts. Is the new classification novel in all senses? Is the new classification only partially novel, with many landforms matching existing classifications? This article discusses how to assess the value of clusters generated by machine learning tools as applied to the analysis of Mars' surface.

BACKGROUND ON CLUSTER VALIDATION

We narrow our discussion to patterns in the form of clusters as produced by a clustering algorithm (a form of unsupervised learning). The goal of a clustering algorithm is to partition the data such that the average distance between objects in the same cluster (i.e., the average intra-distance) is significantly less than the distance between objects in different clusters (i.e., the average inter-distance). The goal is to discover how data objects gather into natural groups (Duda at el., 2001; Bishop, 2006). The application of clustering algorithms can be followed by a post-processing step, also known as cluster validation; this step is commonly employed to assess the quality and meaning of the resulting clusters (Theodoridis and Koutroumbas, 2003).

Cluster validation plays a key role in assessing the value of the output of a clustering algorithm by computing statistics over the clustering structure. Cluster validation is called *internal* when statistics are devised

to capture the quality of the induced clusters using the available data objects only (Krishnapuran et al., 1995; Theodoridis and Koutroumbas, 2003). As an example, one can measure the quality of the resulting clusters by assessing the degree of compactness of the clusters, or the degree of separation between clusters.

On the other hand, if the validation is performed by gathering statistics comparing the induced clusters against an external and independent classification of objects, the validation is called *external*. In the context of planetary science, for example, a collection of sites on a planet constitutes a set of objects that are classified manually by domain experts (geologists) on the basis of their geological properties. In the case of planet Mars, the resultant division of sites into the so-called *geological units* represents an external classification. A clustering algorithm that is invoked to group sites into different clusters can be compared to the existing set of geological units to determine the novelty of the resulting clusters.

Current approaches to external cluster validation are based on the assumption that an understanding of the output of the clustering algorithm can be achieved by finding a resemblance of the clusters with existing classes (Dom, 2001). Such narrow assumption precludes alternative interpretations; in some scenarios high-quality clusters are considered novel if they do not resemble existing classes. After all, a large separation between clusters and classes can serve as clear evidence of cluster novelty (Cheeseman and Stutz, 1996); on the other hand, finding clusters resembling existing classes serves to confirm existing theories of data distributions. Both types of interpretations are legitimate; the value of new clusters is ultimately decided by domain experts after careful interpretation of the distribution of new clusters and existing classes.

In summary, most traditional metrics for external cluster validation output a single value indicating the degree of match between the partition induced by the known classes and the one induced by the clusters. We claim this is the wrong approach to validate patterns output by a data-analysis technique. By averaging the degree of match across all classes and clusters, traditional metrics fail to identify the potential value of individual clusters.

CLUSTER VALIDATION IN MACHINE LEARNING

The question of how to validate clusters appropriately without running the risk of missing crucial information can be answered by avoiding any form of averaging or smoothing approach; one should refrain from computing an average of the degree of cluster similarity with respect to external classes. Instead, we claim, one should compute the distance between each individual cluster and its most similar external class; such comparison can then be used by the domain expert for an informed cluster-quality assessment.

Traditional Approaches to Cluster Validation

More formally, the problem of assessing the degree of match between the set **C** of predefined classes and the set **K** of new clusters is traditionally performed by evaluating a metric where high values indicate a high similarity between classes and clusters. For example, one type of statistical metric is defined in terms of a **2x2** table where each entry $\mathbf{E}_{ij}$, $i,j \in \{1,2\}$, counts the number of object pairs that agree or disagree with the class and cluster to which they belong; $\mathbf{E}_{11}$ corresponds to the number of object pairs that belong to the same class and cluster, $\mathbf{E}_{12}$ corresponds to same class and different cluster, $\mathbf{E}_{21}$ corresponds to different class and same cluster, and $\mathbf{E}_{22}$ corresponds to different class and different cluster. Entries along the diagonal denote the number of object pairs contributing to high similarity between classes and clusters, whereas elements outside the diagonal contribute to a high degree of dissimilarity. A common family of statistics used as metrics simply average correctly classified class-cluster pairs by a function of all possible pairs. A popular similarity metric is Rand's metric (Theodoridis and Koutroumbas, 2003):

$$(\mathbf{E}_{11} + \mathbf{E}_{22}) / (\mathbf{E}_{11} + \mathbf{E}_{12} + \mathbf{E}_{21} + \mathbf{E}_{22})$$

Other metrics are defined as follows:

Jaccard:

$$\mathbf{E}_{11} / (\mathbf{E}_{11} + \mathbf{E}_{12} + \mathbf{E}_{21})$$

Fowlkes and Mallows:

$$\mathbf{E}_{11} / [(\mathbf{E}_{11} + \mathbf{E}_{12}) (\mathbf{E}_{21} + \mathbf{E}_{22})]^{1/2}$$

A different approach is to work on a contingency table **M**, defined as a matrix of size **m**x**n** where each row corresponds to an external class and each column to a cluster. An entry $\mathbf{M}_{ij}$ indicates the number of objects covered by class $\mathbf{C}_i$ and cluster $\mathbf{K}_j$. Using **M**, the similarity between **C** and **K** can be quantified into a single number in several forms (Kanungo et al., 1996; Vaithyanathan and Dom, 2000).

Limitations of Current Metrics

In practice, a quantification of the similarity between sets of classes and clusters is of limited value; we claim any potential discovery provided by the clustering algorithm is only identifiable by analyzing the meaning of each cluster individually. As an illustration, assume two clusters that bear a strong resemblance with two real classes, with a small novel third cluster bearing no resemblance to any class. Averaging the similarity between clusters and classes altogether disregards the potential discovery carried by the third cluster. If the third cluster is small enough, most metrics would indicate a high degree of class-cluster similarity.

In addition, even when in principle one could analyze the entries of a contingency matrix to identify clusters having little overlap with existing classes, such information cannot be used in estimating the intersection of the true probability models from which the objects are drawn. This is because the lack of a probabilistic model in the representation of data distributions precludes estimating the extent of the intersection of a class-cluster pair; probabilistic expectations can differ significantly from actual counts because the probabilistic model introduces substantial a priori information.

Proposed Approach to Cluster Validation

We show our approach to cluster validation in the context of the analysis of Mars' surface (Vilalta et al., 2007). The current qualitative means of classifying Martian surfaces is by assigning each site to what is called a *geological unit*. One shortcoming of such classification is that it cannot be automated because it is normally assigned subjectively by a domain expert. On the other hand, an automated classification of Martian surfaces is possible using digital topography data. Martian topography data is currently available from the Mars Orbiter Laser Altimeter (MOLA) instrument (Smith et al., 2003). This data can be used to construct a digital elevation model (DEM) of a site on Mars' surface. The DEM is a regular grid of cells with assigned elevation values. Such grid can be processed to generate a dataset of feature vectors (each vector component stands as a topographic feature). The resulting training data set can be used as input to a clustering algorithm.

If the clustering algorithm produces **N** clusters, and there exists **M** externals classes (in our case study **M=16** sixteen Martian geological units), we advocate validating the quality of the clusters by computing the degree of overlap between each cluster and each of the existing external classes. In essence, one can calculate an **NxM** matrix of distances between the clusters and classes. This approach to pattern validation enables us to assess the value of each cluster independently of the rest, and can lead to important discoveries.

A practical implementation for this type of validation is as follows. One can model each cluster and each class as a multivariate Gaussian distribution; the degree of separation between both distributions can then be computed using an information-theoretic measure known as relative entropy or Kullback-Leibler distance (Cover and Thomas, 2006). The separation of two distributions can be simplified if it is done along a single dimension that captures most of the data variability (Vilalta et al., 2007). One possibility is to project all data objects over the vector that lies orthogonal to the hyper-plane that maximizes the separation between cluster and class, for example, by using Fisher's Linear Discriminant (Duda, et al., 2001). The resulting degree of overlap can be used as a measure of the similarity of class and cluster. As mentioned before, this approach enables us to assess clusters individually. In some cases a large separation (low overlap) may indicate domain novelty, whereas high overlap or similarity may serve to reinforce current classification schemes.

A CASE STUDY IN PLANETARY SCIENCE

In the analysis of Mars' surface, our clustering algorithm produced **N=9** clusters. Using the method for external cluster assessment explained above, we were able to determine that partitioning the dataset of Martian sites on the basis of the resulting clusters produced a novel classification that does not match the traditional classification based on (**M=16**) geological units. We could conclude this by analyzing each cluster individually,

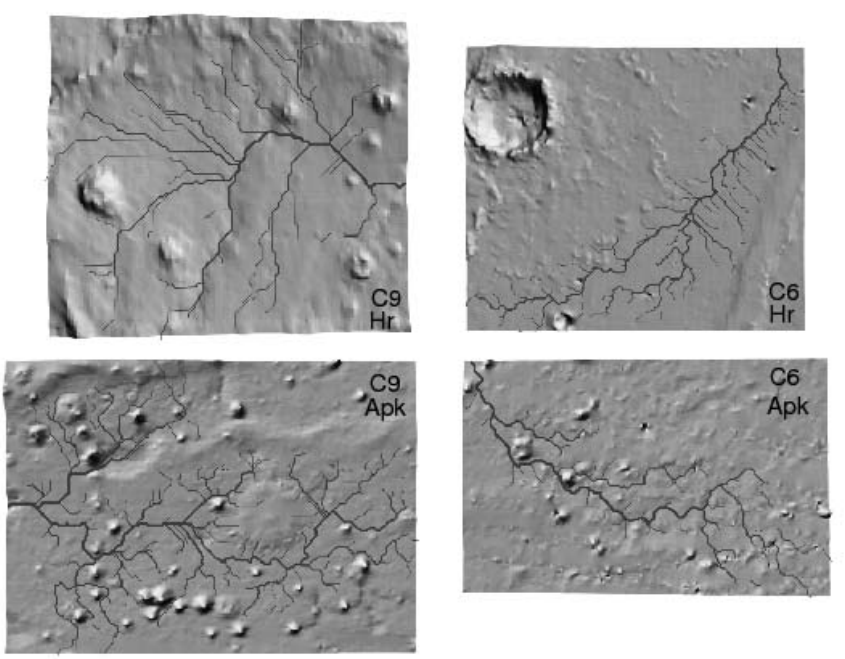
observing no close resemblance with existing classes (in addition our methodology indicates which clusters are more dissimilar than others when compared to the set of classes).

Figure 1 shows an example of the difference between patterns found by clusters and those pertaining to more traditional geomorphic attributes. Four Martian surfaces are shown in a **2x2** matrix arrangement. Surfaces in the same row belong to the same geological unit, whereas surfaces in the same column belong to the same cluster. The top two sites show two surfaces from a geological unit that is described as "ridged plains, moderately cratered, marked by long ridges." These features can indeed be seen in the two surfaces. Despite such texture similarity they exhibit very different shape for their drainage (blue river-like features). The bottom two sites show two surfaces from a geological unit described as "smooth plain with conical hills or knobs." Again, it is easy to see the similarity between these two surfaces based on that description. Nevertheless, the two terrains have drainage with markedly different character. On the basis of the cluster distinction, these four surfaces could be divided vertically instead of horizontally. Such division corresponds to our cluster partition where the drainage is similar for sites within the same cluster. Our methodology facilitates this type of comparisons because clusters are compared individually to similar classes. Domain experts can then provide a scientific explanation of the new data categorization by focusing on particular differences or similarities between specific class-cluster pairs.

FUTURE TRENDS

Future trends include assessing the value of clusters obtained with alternative clustering algorithms (other than probabilistic algorithms). Another trend is to devise modeling techniques for the external class distribution (e.g., as a mixture of Gaussian models). Finally, one line of research is to design clustering algorithms that search for clusters in a direction that maximizes a metric of relevance or *interestingness* as dictated by an external classification scheme. Specifically, a clustering algorithm can be set to optimize a metric that rewards clusters exhibiting little (conversely strong) resemblance to existing classes.

Figure 1. Four Martian surfaces that belong to two different geological units (rows), and two different clusters (columns). Drainage networks are drawn on top of the surfaces.



CONCLUSION

Cluster validation looks for methods to assess the value of patterns output by machine learning techniques. In the context of unsupervised learning or clustering, data objects are grouped into new categories that convey potentially meaningful and novel domain interpretations. When the same data objects have been previously framed into a particular classification scheme, the value of each cluster can be assessed by estimating the degree of separation between the cluster and its most similar class. In this document we criticize common approaches to pattern validation where the mechanism consists of computing an average of the degree of similarity between clusters and classes. Instead we advocate an approach to external cluster assessment based on modeling each cluster and class as a probabilistic distribution; the degree of separation between both distributions can then be measured using an information-theoretic approach (e.g., relative entropy or Kullback-Leibler distance). By looking at each cluster individually, one can assess the degree of novelty (large separation to other classes) of each cluster, or instead the degree of validation (close resemblance to other classes) provided by the same cluster.

REFERENCES

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. New York, NY: Springer.

Bue, B. D., & Stepinski, T. F. (2006). Automated Classification of Landforms on Mars, *Computers & Geoscience*, 32(5), 604-614.

Cheeseman, P., & Stutz, J. (1996). Bayesian Classification (AutoClass): Theory and Results. In Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., & Uthurusamy, R. (Ed.), *Advances in Knowledge Discovery and Data Mining*, AAAI Press/MIT Press.

Cover, T. M., & Thomas, J. A. (2006). *Elements of Information Theory*. John Wiley & Sons.

Dom, B. (2001). *An Information-Theoretic External Cluster-Validity Measure* (Tech. Research Rep. No. 10219). IBM T.J. Watson Research Center.

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern Classification*. Canada: John Wiley, 2nd Edition.

Kanungo, T., Dom., B., Niblack, W., & Steele, D. (1996). A Fast Algorithm for MDL-based Multi-Band Image Segmentation. In: Sanz, J. (ed) Image Technology. Springer-Verlag, Berlin.

Krishnapuran, R., Frigui, H., & Nasraoui, O. (1995). Fussy and Possibilistic Shell Clustering Algorithms and Their Application to Boundary Detection and Surface Approximation, Part II, *IEEE Transactions on Fuzzy Systems*, 3(1), 44-60.

Smith, D., Neumann, G., Arvidson, R. E., Guinness, E. A., & Slavney, S. (2003). Mars Global Surveyor Laser Altimeter Mission Experiment Gridded Data Record, *NASA Planetary Data System, MGS-M-MOLA-5-MEGDR-L3-V1.0.*

Stepinski, T. F., Ghosh, S., & Vilalta, R. (2007). Machine Learning for Automatic Mapping of Planetary Surfaces. *Nineteenth Innovative Applications of Artificial Intelligence Conference*.

Stepinski, T. F., Ghosh, S., & Vilalta, R. (2006). Automatic Recognition of Landforms on Mars Using Terrain Segmentation and Classification. *International Conference on Discovery Science* (pp. 255-266). LNAI 4265.

Stepinski, T. F., & Vilalta, R. (2005). Digital Topography Models for Martian Surfaces, *IEEE Geoscience and Remote Sensing Letters*, 2(3), 260-264.

Theodoridis, S. & Koutroumbas, K. (2003). *Pattern Recognition.* San Diego, CA: Academic Press.

Vaithyanathan, S., & Dom., B. (2000). Model Selection in Unsupervised Learning with Applications to Document Clustering. Proceedings of the 6th International Conference on Machine Learning, Stanford University, CA.

Vilalta, R., Stepinski, T., & Achari, M. (2007). An Efficient Approach to External Cluster Assessment with an Application to Martian Topography. *Data Mining and Knowledge Discovery Journal*, 14, 1-23.

Wilhelms, D. E. (1990). Geologic Mapping. In: Greeley, R., Batson, R. (Eds.), *Planetary Mapping.* Cambridge University Press, Cambridge, UK, pp 209-244.

KEY TERMS

Cluster Validation: A post-processing step after clustering used to assess the value of the resulting clusters.

Digital Elevation Model: A digital elevation model (DEM) is a regular grid of cells with assigned elevation values. It characterizes a particular site based on the shape of the terrain.

External Cluster Validation: Cluster validation is called *external* if the validation is performed by gathering statistics comparing the induced clusters against an external and independent classification of objects.

Fisher's Linear Discriminant: Fisher's linear discriminant finds a hyperplane that separates two data clusters by searching for a normal vector **w** that maximizes the separation between clusters when the data is projected over **w**.

Internal Cluster Validation: Cluster validation is called *internal* when statistics are devised to capture the quality of the induced clusters using the available data objects only.

MOLA Instrument: MOLA is the Mars Orbiter Laser Altimeter; it is an instrument attached to the Mars Global Surveyor spacecraft sent to Mars in 1996.

Relative Entropy: The relative entropy or Kullback-Leibler distance between two probability mass functions $\mathbf{p(x)}$ and $\mathbf{q(x)}$ is defined as $\mathbf{D(p \,||\, q) = \Sigma_x\, p(x) \log [p(x)/q(x)]}$ (Cover & Thomas, 2006).

Clustering Analysis of Data with High Dimensionality

Athman Bouguettaya
CSIRO ICT Center, Australia

Qi Yu
Virginia Tech, USA

INTRODUCTION

Clustering analysis has been widely applied in diverse fields such as data mining, access structures, knowledge discovery, software engineering, organization of information systems, and machine learning. The main objective of cluster analysis is to create groups of objects based on the degree of their association (Kaufman & Rousseeuw, 1990; Romesburg, 1990).

There are two major categories of clustering algorithms with respect to the output structure: partitional and hierarchical (Romesburg, 1990). K-means is a representative of the partitional algorithms. The output of this algorithm is a flat structure of clusters. The K-means is a very attractive algorithm because of its simplicity and efficiency, which make it one of the favorite choices to handle large datasets. On the flip side, it has a dependency on the initial choice of number of clusters. This choice may not be optimal, as it should be made in the very beginning, when there may not exist an informal expectation of what the number of natural clusters would be. Hierarchical clustering algorithms produce a hierarchical structure often presented graphically as a dendrogram. There are two main types of hierarchical algorithms: agglomerative and divisive. The agglomerative method uses a bottom-up approach, i.e., starts with the individual objects, each considered to be in its own cluster, and then merges the clusters until the desired number of clusters is achieved. The divisive method uses the opposite approach, i.e., starts with all objects in one cluster and divides them into separate clusters. The clusters form a tree with each higher level showing higher degree of dissimilarity. The height of the merging point in the tree represents the similarity distance at which the objects merge in one cluster. The agglomerative algorithms are usually able to generate high-quality clusters but suffer a high computational complexity compared with divisive algorithms.

In this paper, we focus on investigating the behavior of agglomerative hierarchical algorithms. We further divide these algorithms into two major categories: group based and single-object based clustering methods. Typical examples for the former category include Unweighted Pair-Group using Arithmetic averages (UPGMA), Centroid Linkage, and WARDS, etc. Single LINKage (SLINK) clustering and Complete LINKage clustering (CLINK) fall into the second category. We choose UPGMA and SLINK as the representatives of each category and the comparison of these two representative techniques could also reflect some similarity and difference between these two sets of clustering methods. The study examines three key issues for clustering analysis: (1) the computation of the degree of association between different objects; (2) the designation of an acceptable criterion to evaluate how good and/or successful a clustering method is; and (3) the adaptability of the clustering method used under different statistical distributions of data including random, skewed, concentrated around certain regions, etc. Two different statistical distributions are used to express how data objects are drawn from a 50-dimensional space. This also differentiates our work from some previous ones, where a limited number of dimensions for data features (typically up to three) are considered (Bouguettaya, 1996; Bouguettaya & LeViet, 1998). In addition, three types of distances are used to compare the resultant clustering trees: Euclidean, Canberra Metric, and Bray-Curtis distances. The results of an exhaustive set of experiments that involve data derived from 50-dimensional space are presented. These experiments indicate a surprisingly high level of similarity between the two clustering techniques under most combinations of parameter settings.

The remainder of this paper is organized as follows. Section 2 discusses the clustering techniques used in our evaluation and describes the various distributions

used to derive our experimental data. Section 3 outlines the experimental methodology and Section 4 presents a summary of our results. Finally, concluding remarks are drawn in Section 5.

BACKGROUND

In this section, we outline a set of key elements for conducting clustering analysis. These include *distances of similarity, coefficients of correlation, clustering methods, and statistical distributions of data objects.* In what follows, we will give a detailed discussion of each of these elements. Finally, we present a general algorithm, which outlines the procedure of constructing clustering in our study.

Distances of Similarity

To cluster data objects in a database system or in any other environment, some means of quantifying the degree of associations between items is needed. This can be a measure of distances or similarities. There are a number of similarity measures available and the choice may have an effect on the results obtained. Multi-dimensional objects may use relative or normalized weight to convert their distance to an arbitrary scale so they can be compared. Once the objects are defined in the same measurement space as the points, it is then possible to compute the degree of similarity. In this respect, the smaller the distance the more similar two objects are. The most popular choice in computing distance is the Euclidean distance with:

$$d(i, j) = \sqrt{(x_{i_1} - x_{j_1})^2 + (x_{i_2} - x_{j_2})^2 + ... + (x_{i_n} - x_{j_n})^2} \quad (1)$$

Euclidean distance belongs to the family of Minkowski's distances, which is defined as

$$d(i, j) = (| x_{i_1} - x_{j_1} |^m + | x_{i_2} - x_{j_2} |^m + ... + | x_{i_n} - x_{j_n} |^m)^{\frac{1}{m}} \quad (2)$$

When m = 2, Minkowski's distance becomes Euclidean distance. Another widely used distance, called Manhattan distance, is also a special case of Minkowski's distance (when m is set to 1).

In addition to Euclidean distance, we also use another two types of distances to investigate how this element could affect clustering analysis: *Canberra Metric* and *Bray-Curtis distances*. Canberra Metric distance, a(i, j), has a range between 0.0 and 1.0. The data objects i and j are identical when a(i, j) takes value 0.0. Specifically, a(i, j) is defined as:

$$a(i, j) = \frac{1}{n}\left(\frac{| x_{i_1} - x_{j_1} |}{(x_{i_1} + x_{j_1})} + \frac{| x_{i_2} - x_{j_2} |}{(x_{i_2} + x_{j_2})} + ... + \frac{| x_{i_n} - x_{j_n} |}{(x_{i_n} + x_{j_n})}\right) \quad (3)$$

Similarly, Bray-Curtis distance, b(i, j), also has values ranged from 0.0 to 1.0. The value 0.0 indicates the maximum similarity between two data objects. b(i, j) is defined as:

$$b(i, j) = \frac{| x_{i_1} - x_{j_1} | + | x_{i_2} - x_{j_2} | + ... + | x_{i_n} - x_{j_n} |}{(x_{i_1} + x_{j_1}) + (x_{i_2} + x_{j_2}) + ... + (x_{i_n} + x_{j_n})} \quad (4)$$

Coefficients of Correlation

Coefficients of correlation are the measurements that describe the strength of the relationship between two variables X and Y. It essentially answers the question *"how similar are X and Y?"*. In our study, coefficients of correlation will be used to compare outcomes (i.e., hierarchical trees) of different clustering techniques. The values of the coefficients of correlation range from 0 to 1 where the value 0 points to *no similarity* and the value 1 points *high similarity*. The coefficient of correlation is used to find the similarity among (clustering) objects. The correlation r of two random variables X and Y where: X = (x_1, x_2, x_3, . . . , x_n) and Y = (y_1, y_2, y_3, . . . , y_n) is given by the formula:

$$r = \frac{| E(X,Y) - E(X) \times E(Y) |}{\sqrt{(E(X^2) - E^2(X)}\sqrt{(E(Y^2) - E^2(Y)}} \quad (5)$$

where

$E(X) = (\sum_{i=1}^{n} x_i)/n$,

$E(Y) = (\sum_{i=1}^{n} y_i)/n$, and

$E(X,Y) = (\sum_{i=1}^{n} x_i \times y_i)/n.$

Clustering Methods

Clustering methods can be classified as partitioning and hierarchical methods according to the type of the group structures they produce. Here, we focus on hierarchical methods which work in a bottom–up and are appropriate for data-intensive environment (Zupan, 1982). The algorithm proceeds by performing a series of successive fusion. This produces a nested data set in which pairs of items or clusters are successively linked until every item in the data set is linked to form one cluster, known also as hierarchical tree. This tree is often presented as a *dendrogram*, in which pairwise couplings of the objects in the data set are shown and the length of the branches (vertices) or the value of the similarity is expressed numerically. In this study, we focus on two methods that enjoy wide usage: Unweighted Pair-Group using Arithmetic Averages (UPGMA) and Single LINKage (SLINK) clustering. UPGMA is based on comparing groups of data objects, whereas SLINK is based on single object comparison.

- *Unweighted Pair-Group using Arithmetic Averages*: This method uses the average values pair-wise distance, denoted $D_{X,Y}$, within each participating cluster to determine similarity. All participating objects contribute to inter-cluster similarity. This methods is also known as one of "average linkage" clustering methods (Lance and Williams 1967; Everitt 1977; Romesburg 1990). The distance between two clusters is

$$D_{X,Y} = \frac{\sum D_{x,y}}{n_X \times n_Y} \qquad (6)$$

  where X and Y are two clusters, x and y are objects from X and Y , Dx,y is the distance between x and y, and n_X and n_Y are the respective sizes of the clusters.
- *Single LINKage Clustering:* This method is also known as the nearest neighbor method. The distance between two clusters is the smallest distance of all distances between two items *(x, y)*, denoted $d_{i,j}$, such that x is a member of a cluster and y is a member of another cluster. The distance $d_{I,J}$ is computed as follows:

$$d_{I,J} = min\{d_{i,j}\} \text{ with } i \in I, j \in J \qquad (7)$$

  where (I, J) are clusters, and (i, j) are objects in the corresponding clusters. This method is the simplest among all clustering methods. It has some attractive theoretical properties (Jardine and Sibson 1971). However, it tends to form long or chaining clusters. This method may not be very suitable for objects concentrated around some centers in the measurement space.

Statistical Distributions

As mentioned above, objects that participate in the clustering process are randomly selected from a designated area (i.e., [0, 1] × [0, 1] × [0, 1]...). There are several random distributions. We chose two of them that closely model real world distributions (Zupan, 1982). Our aim is to examine whether clustering is dependent on the way objects are generated. These two distributions are uniform and Zipf's distributions. Below, we describe these statistical distributions in terms of density functions.

- *Uniform Distribution* whose density function is $f(x) = 1$ for all x in $0 \le x \le 1$.
- *Zipf's Distribution* whose density function is $f(x) = C/x^{\alpha}$ for x in $1 \le x \le N$ where C is the normalization constant (i.e., $\sum f(x) = 1$).

A General Algorithm

Before the grouping commences, objects following the chosen probabilistic guidelines are generated. In this paper, each dimension of the objects are randomly drawn from the interval [0, 1]. Subsequently, the objects are compared to each other by computing their distances. The distance used in assessing the similarity between two clusters is called the similarity coefficient. This is not to be confused with *coefficient of correlations* as the latter are used to compare outcomes (i.e., hierarchical trees) of the clustering process. The way objects and clusters of objects coalesce together to form larger clusters varies with the approach used. Below, we outline a generic algorithm that is applicable to all clustering methods (initially, every cluster consists of exactly one object).

1. Create all possible cluster formulations from the existing ones.

2. For each such candidate compute its corresponding similarity coefficient.
3. Find out the minimum of all similarity coefficients and then join the corresponding clusters.
4. If the number of clusters is not equal to one (i.e., not all clusters have coalesced into one entity), then go to (1).

Essentially, the algorithm consists of two phases: the first phase records the similarity coefficients.

The second phase computes the minimum coefficient and then performs the clustering.

There is a case when using average-based methods ambiguity may arise. For instance, let us suppose that when performing Step 1 (of the above algorithmic skeleton), three successive clusters are to be joined. All these three clusters have the same minimum similarity value. When performing Step 2, the first two clusters are joined. However, when computing the similarity coefficient between this new cluster and the third cluster, the similarity coefficient value may now be different from the minimum value. The question at this stage is what the next step should be. There are essentially two options:

- Continue by joining clusters using a recomputation of the similarity coefficient every time we find ourselves in Step 2, or
- Join all those clusters that have the same similarity coefficient at once and do not recompute the similarity in Step 2.

In general, there is no evidence that one is better than the other (Romesburg, 1990). For our study, we selected the first alternative.

MAJOR FOCUS

The sizes of data objects presented in this study range from 50 to 400. Data objects are drawn from a 50-dimensional space. The value of each attribute value ranges from 0 and 1 inclusive. Each experiment goes through the following steps that result in a clustering tree.

- Generate lists of objects with a statistical distribution method (Uniform or Zipf's).
- Carry out the clustering process with a clustering method (UPGMA or SLINK).
- Calculate the coefficient of correlation for each clustering method.

For statistical validation, each experiment is repeated 100 times. The correlation coefficient is used as the main vehicle for comparing two trees obtained from lists of objects. The notion of distance used in the computation of the correlation coefficients could be realized in three ways: Euclidean distance (linear distance), Canberra Metric distance, and Bray-Curtis distance. Once a distance type is chosen, we may proceed with the computation of the correlation coefficient. This is accomplished by first selecting a pair of identifiers (two objects) from a list (linearized tree) and calculating their distance and then by selecting the pair of identifiers from the second list (linearized tree) and computing their distance. We repeat the same process for all remaining pairs in the second list.

There are numerous families of correlation coefficients that could be examined. This is due to the fact that various parameters are involved in the process of evaluating clustering of objects in the two-dimensional space. More specifically, the clustering method is one parameter (i.e., UPGMA and SLINK); the method of computing the distances is another one (i.e., linear, Canberra Metric, and Bray-Curtis); and finally, the distribution followed by the data objects (i.e., uniform and Zipf's) is a third parameter. In total, there are 12 (e.g., 2*3*2=12) possible ways to compute correlation coefficients for any two lists of objects. In addition, the dimensional space and the input size may also have a direct influence on the clustering. This determines what kinds of data are to be compared and what their sizes are.

We have identified a number of cases to check the sensitivity of each clustering method with regard to the input data. For every type of coefficient of correlation mentioned above, four types of situations (hence, four coefficients of correlation) have been isolated. All these types of situations are representative of a wide range of practical settings (Bouguettaya, 1996) and can help us understand the major factors that influence the choice of a clustering method (Jardine & Sibson, 1971; Kaufman & Rousseeuw, 1990; Tsangaris & Naughton, 1992).

We partition these settings in three major groups, represented by three templates or blocks of correlation coefficients.

First Block: the coefficients presented in this set examine the influence of context in how objects are finally clustered. In particular, the correlation coefficients are between:

- Pairs of objects drawn from a set *S* and pairs of objects drawn from the first half of the same set *S*. The first half of S is used before the set is sorted.
- Pairs of objects drawn from the first half of *S*, say S_2, and pairs of objects drawn from the first half of another set *S'*, say S'_2. The two sets are given ascending identifiers after being sorted. The first object of S_2 is given as identifier the number 1 and so is given the first object of S'_2. The second object of S_2 is given as identifier the number 2 and so is given the second object of S'_2 and so on.

Second Block: This set of coefficients determines the influence of the data size. Coefficients of correlation are drawn between:

- Pairs of objects drawn from S and pairs of objects drawn from the union of a set *X* and *S*. The set *X* contains 10% new randomly generated objects.

Third Block: The purpose of this group of coefficients is to determine the relationship that may exist between two lists of two-dimensional objects derived using different distributions. More specifically, the coefficients of correlation are drawn between:

- Pairs of objects drawn from *S* using the uniform distribution and pairs of objects drawn from *S'* using the Zipf's distribution.

In summary, all the above four types of coefficients of correlation are meant to analyze different settings in the course of our evaluation.

Results and Interpretation

We annotate all the figures below using a shorthand notation as follows: UPGMA (U), SLINK (S), Zipf (Z), Uniform (U), Linear distance (L), Canberra Metric distance (A), Bray-Curtis distance (B), Coefficient of correlation (CC), and Standard deviation (SD). For example, the abbreviation ZUL is used to represent the input with the following parameters: Zipf's distribution, UPGMA method, and Linear distance. In what follows, we provide a thorough interpretation of the results.

Figure 1 depicts the first coefficient of correlation for UPGMA and SLINK. The top two graphs depict the coefficient of correlation (CC) and standard deviation (SD) from UPGMA clustering method. In each graph, the three CC curves are above the three SD curves. The top left graph in Figure 1 is based on Zifp's distribution whereas the top right graph is based on uniform distribution. The choice of distribution does not seem to have any influence in the clustering process. However, the distance type seems to have some impact on the clustering. By examining these two graphs, we notice that the correlations are a bit stronger in the case of the Euclidean distance. Because of the high correlation in all cases depicted in these two graphs,

Figure 1. The first coefficient of correlation

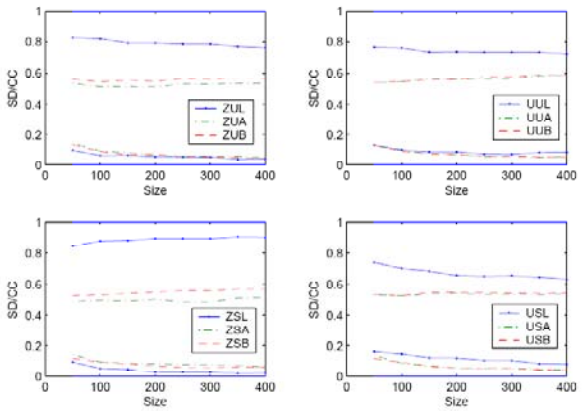
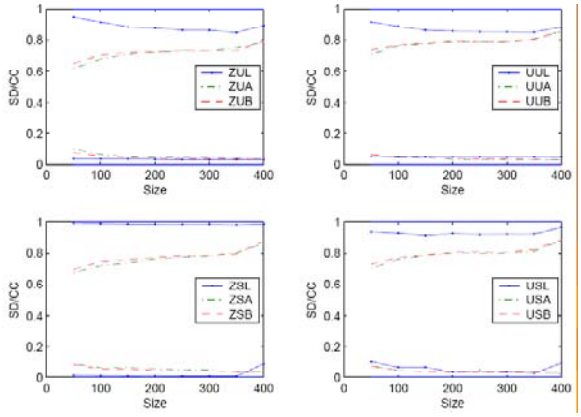


Figure 2. The third coefficient of correlation



the context in this specific case does not seem to make strong significant difference in the clustering process. However, Canberra Metric and Bray-Curtis distances seem to be more sensitive to the context change in this case than Euclidian distance. The three curves in the bottom of these two graphs show the standard deviation of UPGMA method. The low value justifies the stability of this method. If we look at bottom two graphs in Figure 1, which depict the clustering possibilities using SLINK method, the results are very similar. This strengthens the conjecture that no matter what the context of the clustering is, the outcome is almost invariably the same.

The third coefficient of correlation checks the influence of the data size on clustering. They are shown in Figure 2. There are no significant differences when using these two clustering methods respectively. Similar to the situation of the first coefficient of correlation, Euclidian distance generates stronger correlation than the other two types of distances. This shows that Canberra Metric and Bray-Curtis distances seem to be more sensitive to data size change than Euclidian distance. The high correlation and low standard deviation in all cases also indicate that SLINK and UPGMA are relatively stable for multidimensional objects. The perturbation represented by the insertion of new data objects does not seem to have any significant effect on the clustering process.

The experiments regarding the second coefficient of correlation show some surprising results. The previous study results using low dimensionality show above average correlation. In contrast, in this study as shown in Figure 3, the correlation values are low. It is a strong indication that in high dimensions, almost no correlation exists between the two sets of objects. In the case of the first coefficient of correlation, the data objects are drawn from the same initial set. However, the second coefficient of correlation draws the values from different initial sets. The difference with the previous study is that we now randomly select one dimension from multiple dimensions as our reference dimension. As a result, the order of data points is dependent on the value of the selected dimension. The two clustering methods have very similar performance on different types of distances. Since the coefficients of correlation have very small values, the CC curves and SD curves would be mixed with each other. Therefore, we separate CC and SD curves by drawing them in different graphs. This is also applied in the final coefficient of correlation.

The final coefficient of correlation is used to check the influence of the statistical distributions on the clustering process. Figure 4 shows the result of the experiments. It is quite obvious that almost no correlation can be drawn between the two differently generated lists of objects. Simply put, the clustering process is influenced by the statistical distribution used to generate objects.

In order to validate our interpretation of the results, we ran the same experiments for different input size (50-400). The results show the same type of behavior over all these input sizes. The fact that the slope value is close to 0 is the most obvious evidence. Therefore, the input size has essentially no role in the clustering process.

Figure 3. The second coefficient of correlation

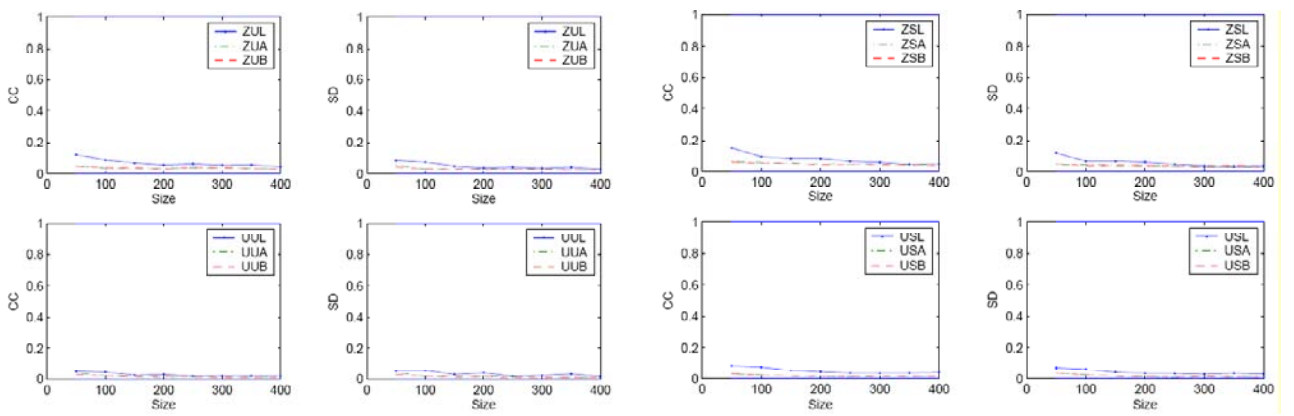


Figure 4. The fourth coefficient of correlation

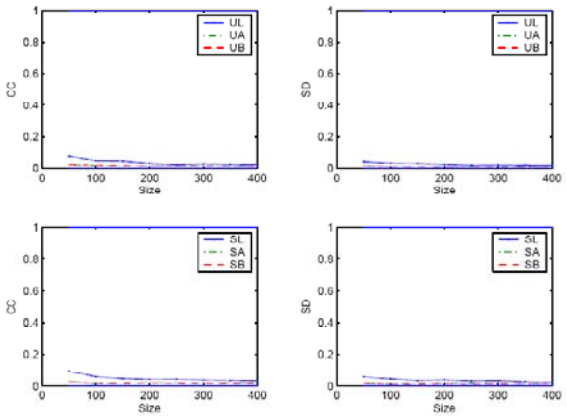
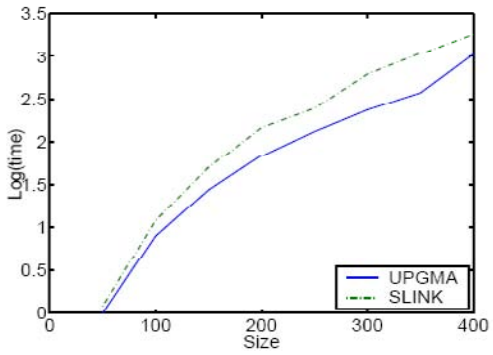


Figure 5. Processing time of UPGMA and SLINK



As a final set of measurements, we also recorded the computation time for various methods. Figure 5 shows the average logarithmic computation time of all experiments varying the dimensions. The considered data input sizes are 50, 100, 150, 200, 250, 300, 350, and 400. The time unit is in (logarithmic) seconds. We can see that UPGMA is slightly more efficient than SLINK. It is interesting to note that both methods exhibit the same computational behavior when varying the dimensions and data input size.

The results obtained in this study confirm several findings from previous studies focusing on low-dimension data objects (1-D, 2-D, and 3-D) (Bouguettaya, 1996). Like previous studies, this set of experiments strongly indicate that there is a basic inherent way for data objects to cluster and independently from the clustering method used. Especially, the dimensionality does not play a significant role in the clustering of data objects. This is compelling evidence that clustering methods in general do not seem to influence the outcome of the clustering process. It is worth indicating that the experiments presented in this paper were computationally intensive. We used 12 Sun Workstations running Solaris 8.0. The complete set of experiments for each clustering methods took on average 5 days to complete.

FUTURE TRENDS

Clustering techniques are usually computing and resource hungry (Romesburg, 1990; Jain, Murty, & Flynn, 1999). In previous research, sample datasets were limited in size because of the high computational requirements of the clustering methods. It has been even more problematic when multidimensional data is analyzed. One research direction has been the investigation of efficient approaches to allow larger datasets to be analyzed. One key requirement is the ability of these techniques to produce good quality clusters while running efficiently and with reasonable memory requirements. Robust algorithms such as hierarchical techniques, that work well with different data types and produce adequate clusters, are quite often not efficient (Jain, Murty et al. 1999). Other algorithms, like partitional techniques, are more efficient but have more limited applicability (Jain, Murty, & Flynn, 1999). For example, some of the most popular clustering algorithms such as SLINK, UPGMA fall in the first category (hierarchical), while others such as K-means fall in this second category (partitional). While SLINK and UPGMA can produce good results, their time complexity is O(n2logn), where n is the size of the dataset. K-means runs in time linear to the number of input data objects but has a number of limitations such the dependency of the data input order, i.e., it is better suited for isotropic input data (Bradley & Fayyad 1998). As a result, the idea of combining the best of both worlds needs to be explored (Gaddam, Phoha, Kiran, & Balagani, 2007; Lin & Chen, 2005). The idea of this new approach is the ability to combine hierarchical and partitional algorithms into a two-stage method so that individual algorithm's strengths are leveraged and the overall drawbacks are minimized.

CONCLUSION

In this study, we analyzed two clustering methods, UPGMA and SLINK. We ran a wide array of experiments to test the stability of sensibility of the clustering methods for high dimensional data objects. The obtained results strengthen previous study result considering low dimensional data objects. The results seem to suggest that if the data is drawn from the same context, then any clustering methods will yield approximately the same highly correlated outcome. The Canberra Metric and Bray-Curtis distances seem to be more sensitive to the context and data size change than the Euclidian distance. When the contexts are different, the results are the same for the two clustering techniques. However, in this case the correlation values are quite low. This is indicative of the significance of choosing different contexts for clustering. The other factors, including the choice of a clustering technique, did not have any bearing on the outcome.

REFERENCES

Bouguettaya, A. (1996). "On-line Clustering." IEEE Transactions on Knowledge and Data Engineering 8(2).

Bouguettaya, A. LeViet, Q. & Delis, A. (1999). Data education. Encyclopedia of Electrical and Electronics Engineering. New York, NY, John Wiley and Sons.

Bradley, P. S. & Fayyad, U. M. (1998). "Refining initial points for K-Means clustering." Proc. 15th International Conf. on Machine Learning.

Everitt, B. (1977). Cluster Analysis. Yorkshire, England, Heinemann Educational Books.

Gaddam, S. R., Phoha, V. V., Kiran, & Balagani, S. (2007), "K-Means+ID3: A Novel Method for Supervised Anomaly Detection by Cascading K-Means Clustering and ID3 Decision Tree Learning Methods", IEEE Transactions on Knowledge and Data Engineering 19(3).

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). "Data clustering: a review." ACM Computing Surveys 31(3)): 264-323.

Jardine, N. & Sibson, R. (1971). Mathematical Taxonomy. London, John Wiley & Sons.

Kaufman, L. & Rousseeuw, P. J. (1990). Finding Groups in Data, an Introduction to Cluster Analysis. London. United Kingdom, John Wiley & Sons.

Lance, G. N. & Williams, W. T. (1967). "A General Theory for Classification Sorting Strategy." The Computer Journal 9(5): 373-386.

Lin, C. & Chen, M. (2005). "Combining Partitional and Hierarchical Algorithms for Robust and Efficient Data Clustering with Cohesion Self-Merging." IEEE Transactions on Knowledge and Data Engineering 17(2).

Romesburg, H. C. (1990). Cluster Analysis for Researchers. Malabar, FL, Krieger Publishing Company.

Tsangaris, M. & Naughton, J. F. (1992). "On the Performance of Object Clustering Techniques." Proceedings of the International Conference on the Management of Data (SIGMOD).

Zupan, J. (1982). Clustering of Large Data Sets. Letchworth, England, Research Studies Press.

KEY TERMS

Coefficients of Correlation: The measurements that describe the strength of the relationship between two variables X and Y.

Data Clustering: Partitioning of a data set into subsets (clusters), so that the data in each subset shares some common features.

Hierarchical Clustering: Building (agglomerative), or breaking up (divisive), a hierarchy of clusters. It finds successive clusters using previously established clusters.

Similarity Coefficient: The distance used in assessing the similarity between two clusters.

Single LINKage (SLINK): A representative of single-object based clustering methods, where only one data object is used for inter-cluster distance computation.

Unweighted Pair-Group using Arithmetic Averages (UPGMA): A representative of group based clustering methods, where all data objects in a cluster will participate in the computation of inter-cluster distances.

Zipf Distribution: A discrete power-law distribution, the most famous example of which is the description of the frequency of words in the English language.

Clustering Categorical Data with k-Modes

Joshua Zhexue Huang
The University of Hong Kong, Hong Kong

INTRODUCTION

A lot of data in real world databases are categorical. For example, gender, profession, position, and hobby of customers are usually defined as categorical attributes in the CUSTOMER table. Each categorical attribute is represented with a small set of unique categorical values such as {Female, Male} for the gender attribute. Unlike numeric data, categorical values are discrete and unordered. Therefore, the clustering algorithms for numeric data cannot be used to cluster categorical data that exists in many real world applications.

In data mining research, much effort has been put on development of new techniques for clustering categorical data (Huang, 1997b; Huang, 1998; Gibson, Kleinberg, & Raghavan, 1998; Ganti, Gehrke, & Ramakrishnan, 1999; Guha, Rastogi, & Shim, 1999; Chaturvedi, Green, Carroll, & Foods, 2001; Barbara, Li, & Couto, 2002; Andritsos, Tsaparas, Miller, & Sevcik, 2003; Li, Ma, & Ogihara, 2004; Chen, & Liu, 2005; Parmar, Wu, & Blackhurst, 2007). The k-modes clustering algorithm (Huang, 1997b; Huang, 1998) is one of the first algorithms for clustering large categorical data. In the past decade, this algorithm has been well studied and widely used in various applications. It is also adopted in commercial software (e.g., Daylight Chemical Information Systems, Inc, http://www.daylight.com/).

BACKGROUND

In data mining, k-means is the mostly used algorithm for clustering data because of its efficiency in clustering very large data. However, the standard k-means clustering process cannot be applied to categorical data due to the Euclidean distance function and use of means to represent cluster centers. To use k-means to cluster categorical data, Ralambondrainy (1995) converted each unique category to a dummy binary attribute and used 0 or 1 to indicate the categorical value either absent or present in a data record. This approach is not suitable for high dimensional categorical data.

The k-modes approach modifies the standard k-means process for clustering categorical data by replacing the Euclidean distance function with the simple matching dissimilarity measure, using modes to represent cluster centers and updating modes with the most frequent categorical values in each of iterations of the clustering process. These modifications guarantee that the clustering process converges to a local minimal result. Since the k-means clustering process is essentially not changed, the efficiency of the clustering process is maintained. The k-modes clustering algorithm was first published and made publicly available in 1997 (Huang, 1997b). An equivalent approach was reported independently in (Chaturvedi, Green, Carroll, & Foods, 2001). The relationship of the two k-modes methods is described in (Huang & Ng, 2003).

In the past decade, a few other methods for clustering categorical data were also proposed, including the dynamic system approach (Gibson, Kleinberg, & Raghavan, 1998), Cactus (Ganti, Gehrke, & Ramakrishnan, 1999), ROCK (Guha, Rastogi, & Shim, 1999), Coolcat (Barbara, Li, & Couto, 2002), and LIMBO (Andritsos, Tsaparas, Miller, & Sevcik, 2003). However, these methods have largely stayed in research stage and not been widely applied to real world applications.

MAIN FOCUS

The k-modes clustering algorithm is an extension to the standard k-means clustering algorithm for clustering categorical data. The major modifications to k-means include distance function, cluster center representation and the iterative clustering process (Huang, 1998).

Distance Function

To calculate the distance (or dissimilarity) between two objects X and Y described by m categorical attributes, the distance function in k-modes is defined as

$$d(X,Y)=\sum_{j=1}^{m}\delta(x_j,y_j) \quad (1)$$

where

$$\delta(x_j,y_j)=\begin{cases}0, & x_j=y_j\\ 1, & x_j\neq y_j.\end{cases} \quad (2)$$

Here, x_j and y_j are the values of attribute j in X and Y. This function is often referred to as simple matching dissimilarity measure or Hemming distance. The larger the number of mismatches of categorical values between X and Y is, the more dissimilar the two objects.

A new distance function for k-modes defines the dissimilarity measure between an object X and a cluster center Z_l as (Ng, Li, Huang, & He, 2007):

$$\phi(x_j,z_j)=\begin{cases}1-\dfrac{n_j^r}{n_l}, & x_j=z_j\\ 1, & x_j\neq z_j\end{cases} \quad (3)$$

where z_j is the categorical value of attribute j in Z_l, n_l is the number of objects in cluster l and n_j^r is the number of objects whose attribute value is r. In this function, when the categorical value of an object is same as the value of cluster center, its distance depends on the frequency of the categorical value in the cluster.

Cluster Modes

In k-modes clustering, the cluster centers are represented by the vectors of modes of categorical attributes. In statistics, the mode of a set of values is the most frequent occurring value. For example, the mode of set {*a*, *b*, *a*, *a*, *c*, *b*} is the most frequent value *a*. There can be more than one mode in a set of values. If a data set has m categorical attributes, the mode vector Z consists of m categorical values ($z_1, z_2, \ldots, z_m$), each being the mode of an attribute. The mode vector of a cluster minimizes the sum of the distances between each object in the cluster and the cluster center (Huang, 1998).

Clustering Process

To cluster a categorical data set $\boldsymbol{X}$ into k clusters, the k-modes clustering process consists of the following steps:

Step 1: Randomly select k unique objects as the initial cluster centers (modes).

Step 2: Calculate the distances between each object and the cluster mode; assign the object to the cluster whose center has the shortest distance to the object; repeat this step until all objects are assigned to clusters.

Step 3: Select a new mode for each cluster and compare it with the previous mode. If different, go back to Step 2; otherwise, stop.

This clustering process minimizes the following k-modes objective function

$$F(U,Z)=\sum_{l=1}^{k}\sum_{i=1}^{n}\sum_{j=1}^{m}u_{i,l}d(x_{i,j},z_{l,j})$$

where $U=[u_{i,j}]$ is an $n\times k$ partition matrix, $Z = \{Z_1, Z_2, \ldots, Z_k\}$ is a set of mode vectors and the distance function $d(.,.)$ is defined as either (2) or (3). Since it is essentially same as k-means, the k-modes clustering algorithm is efficient in clustering large categorical data and also produces locally minimal clustering results.

Fuzzy k-Modes and Other Variants

The fuzzy k-modes clustering algorithm is an extension to k-modes (Huang & Ng, 1999). Instead of assigning each object to one cluster, the fuzzy k-modes clustering algorithm calculates a cluster membership degree value for each object to each cluster. Similar to the fuzzy k-means, this is achieved by introducing the fuzziness factor in the objective function (Huang & Ng, 1999). The fuzzy k-modes clustering algorithm has found new applications in bioinformatics (Thornton-Wells, Moore, & Haines, 2006). It can improve the clustering result whenever the inherent clusters overlap in a data set.

The k-prototypes clustering algorithm combines k-means and k-modes to cluster data with mixed numeric and categorical values (Huang, 1997a). This is achieved

by defining a new distance function that combines both squared Euclidean distance and the simple matching dissimilarity measure as

$$d(X,Y) = \sum_{j=1}^{p}(x_j - y_j)^2 + \gamma \sum_{j=p+1}^{m}\delta(x_j, y_j) \qquad (5)$$

where the first term is the sum of the distances of the p numeric attributes and the second term is the sum of the distances of the (m-p) categorical attributes. Here, γ is a weighting factor to balance the importance of numeric and categorical attributes in the clustering result. In the clustering process, the means and the modes are used to represent the cluster centers for numeric and categorical attributes, respectively.

The W-k-means clustering algorithm was a recent extension that can automatically calculate the attribute weights in the clustering process (Huang, Ng, Rong, & Li, 2005). In W-k-means, a set of weights $V=\{v_1, v_2, \ldots, v_m\}$ is introduced to the objective function as

$$F(U,Z,V) = \sum_{l=1}^{k}\sum_{i}^{n}\sum_{j}^{m} u_{i,l} v_j^{\beta}\, d(x_{i,j}, z_{l,j}) \qquad (6)$$

where v_j is the weight for attribute j and

$$\sum_{j=1}^{m} v_j = 1.$$

β (>1) is a given factor. This function is minimized by the k-means clustering process with an additional step to calculate the attribute weights in each of iterations as

$$v_j = \begin{cases} 0 & \text{if } D_j = 0 \\ \dfrac{1}{\sum_{t=1}^{h}\left(\frac{D_j}{D_t}\right)^{\frac{1}{\beta-1}}} & \text{if } D_j \neq 0 \end{cases} \qquad (7)$$

where

$$D_j = \sum_{l=1}^{k}\sum_{i=1}^{n} u_{i,l} d(x_{i,j}, z_{l,j}) \qquad (8)$$

The weight of an attribute is inversely proportional to the dispersion of its values in the clusters. The less the dispersion is, the more important the attribute. Therefore, the weights can be used to select important attributes in cluster analysis.

FUTURE TRENDS

A simple Web survey with Google can reveal that a lot of k-modes related research is on-going in many organizations. The research directions can be summarized as follows:

1. Methods for selection of initial modes to generate better clustering results (Khan & Kant, 2007),
2. How to determine the number of clusters,
3. New distance functions to consider more factors such as the structure of the categorical attribute values,
4. Subspace clustering of categorical data (Jing, Ng, & Huang, 2007),
5. Clustering validation methods for categorical data, and
6. New application problems.

CONCLUSION

Over a decade development and applications, the k-modes clustering algorithm has become an important data mining technique for categorical data clustering. As the complexity of data and analysis increases in many application domains, further enhancements to the standard k-modes clustering algorithm will continue in the future and new variants will come out to tackle new problems such as automatic discovery of the number of clusters and subspace clustering of categorical data. The k-modes related techniques will continue finding new applications in broad application areas where categorical data is heavily involved. More commercial and research software will adopt the k-modes technique as its function for categorical data clustering.

REFERENCES

Andritsos, P., Tsaparas, P., Miller, R.J., & Sevcik, K.C. (2003). *LIMBO: a scalable algorithm to cluster categorical data* (Tech. Report CSRG-467), University of Toronto, Department. of Computer Science.

Barbara, D. Li, Y., & Couto, J. (2002). Coolcat: an entropy-based algorithm for categorical clustering. In *Proceedings of ACM Conference on Information and Knowledge Management (CIKM)* (pp. 582-589), ACM Press.

Chaturvedi, A., Green, P. E., Carroll, J. D., & Foods, K. (2001). k-modes clustering. *Journal of Classification*, 18(1), 35-55.

Chen, K., & Liu, L. (2005). The 'best k' for entropy-based categorical Clustering, In James Frew (ed.), *Proc of SSDBM05* (pp.253-262), Donald Bren School of Environmental Science and Management, University of California Santa Barbara CA, USA.

Ganti, V., Gehrke, J., & Ramakrishnan, R. (1999). Cactus–clustering categorical data using summaries. In A. Delis, C. Faloutsos, S. Ghandeharizadeh (Eds.), *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 73–83), ACM Press.

Gibson, D., Kleinberg, J., & Raghavan, P. (1998). Clustering categorical data: An approach based on dynamic systems. In A. Gupta, O. Shmueli, & J. Widom (Eds.), *Proceedings of the 24th International Conference on Very Large Databases* (pp. 311 – 323), Morgan Kaufmann.

Guha, S., Rastogi, R., & Shim, K. (1999). ROCK: A robust clustering algorithm for categorical attributes. In *Proceedings of ICDE* (pp. 512-521), IEEE Computer Society Press.

Huang, J. Z. (1997a). Clustering large data sets with mixed numeric and categorical values. In H. Lu, H. Motoda & H. Liu (Eds), *PAKDD'97. KDD: Techniques and Applications* (pp. 21-35). Singapore: World Scientific.

Huang, J. Z. (1997b). *A fast clustering algorithm to cluster very large categorical data sets in data mining*. In R. Ng (Ed). *1997 SIGMOD Workshop on Research Issues on Data Mining and Know ledge Discovery* (pp. 1-8, Tech. Rep 97-07). Vancouver, B.C., Canada: The University of British Columbia, Department of Computer Science.

Huang, J. Z. (1998). Extensions to the k-means algorithm for clustering large data sets with categorical values. *International Journal of Data Mining and Knowledge Discovery*, 2(3), 283-304.

Huang, J. Z., & Ng, K. P. (1999). A fuzzy k-modes algorithm for clustering categorical data. *IEEE Transactions on Fuzzy Systems*, 7(4), 446-452.

Huang, J. Z., & Ng, K. P. (2003). A note on k-modes clustering. *Journal of Classification*, 20(2), 257-261.

Huang, J. Z., Ng, K. P., Rong, H., & Li, Z. (2005). Automated variable weighting in k-means type clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5), 657-668.

Jing, L., Ng, K. P. & Huang, J. Z. (2007). An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data, *IEEE Transactions on Knowledge and Data Engineering*, 19(8), 1026-1041.

Khan, S. S., & Kant, S. (2007). Computation of initial modes for k-modes clustering algorithm using evidence accumulation. In M. M. Veloso (Ed.), Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI-07) (pp. 2784-2789), IJCAI.

Li, T., Ma, S. & Ogihara, M. (2004). Entropy-based criterion in categorical clustering. In C. E. Brodley (Ed.), *Proceeding of International Conference on Machine Learning (ICML)* (pp. 536-543), ACM Press.

Ng, K. P., Li, M., Huang, J. Z., & He, Z. (2007). On the impact of dissimilarity measure in k-modes clustering algorithm, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(3), 503-507.

Parmar, D., Wu, T., & Blackhurst, J. (2007). MMR: An algorithm for clustering categorical data using Rough Set Theory, *Data & Knowledge Engineering*, 63(3), 879-893.

Ralambondrainy, H. (1995). A conceptual version of the k-means algorithm. *Pattern Recognition Letters*, 16(11), 1147-1157.

Thornton-Wells, T. A., Moore, J. H., & Haines, J. L. (2006). Dissecting trait heterogeneity: a comparison of three clustering methods applied to genotypic data, *BMC Bioinformatics*, 7(204).

KEY TERMS

Categorical Data: Data with values taken from a finite set of categories without orders.

Cluster Validation: Process to evaluate a clustering result to determine whether the "true" clusters inherent in the data are found.

Clustering: A process of grouping data objects into clusters according to some similarity measure.

Fuzzy k-Modes: A fuzzy version of k-modes to cluster categorical data.

k-Means: A partitional clustering technique that partitions data objects into *k* clusters by iteratively minimizing the sum of the within cluster dispersions.

k-Modes: A modification of k-means to cluster categorical data.

k-Prototypes: A combination of k-means and k-modes to cluster data with mixed numeric and categorical values.

W-k-Means: An extension to k-prototypes to enable automatic calculation of attribute weights in the clustering process for attribute selection.

Clustering Data in Peer-to-Peer Systems

Mei Li
Microsoft Corporation, USA

Wang-Chien Lee
Pennsylvania State University, USA

INTRODUCTION

With the advances in network communication, many large scale network systems have emerged. Peer-to-peer (P2P) systems, where a large number of nodes self-form into a dynamic information sharing system, are good examples. It is extremely valuable for many P2P applications, such as market analysis, scientific exploration, and smart query answering, to discover the knowledge hidden in this distributed data repository. In this chapter, we focus on clustering, one of the most important data mining tasks, in P2P systems. We outline the challenges and review the start-of-the-art in this area.

Clustering is a data mining technique to group a set of data objects into classes of similar data objects. Data objects within the same class are similar to each other, while data objects across classes are considered as dissimilar. Clustering has a wide range of applications, e.g., pattern recognition, spatial data analysis, custom/market analysis, document classification and access pattern discovery in WWW, etc.

Data mining community have been intensively studying clustering techniques for the last decade. As a result, various clustering algorithms have been proposed. Majority of these proposed algorithms is designed for traditional centralized systems where all data to be clustered resides in (or is transferred to) a central site. However, it is not desirable to transfer all the data from widely spread data sources to a centralized server for clustering in P2P systems. This is due to the following three reasons: 1) there is no central control in P2P systems; 2) transferring all data objects to a central site would incur excessive communication overheads, and 3) participants of P2P systems reside in a collaborating yet competing environment, and thus they may like to expose as little information as possible to other peers for various reasons. In addition, these existing algorithms are designed to minimize disk access cost. In P2P system, the communication cost is a dominating factor. Therefore, we need to reexamine the problem of clustering in P2P systems.

A general idea to perform clustering in P2P systems is to first cluster the local data objects at each peer and then combine the local clustering results to form a global clustering result. Based on this general idea, clustering in P2P systems essentially consists of two steps, i.e., *local clustering* and *cluster assembly*. While local clustering can be done by employing existing clustering techniques, cluster assembly is a nontrivial issue, which concerns *representation model* (what should be communicated among peers) and *communication model* (how peers communicate with each other).

In this chapter, we review three representation models (including two *approximate representation models* and an *exact representation model*) and three communication models (including *flooding-based communication model*, *centralized communication model*, and *hierarchical communication model*).

The rest of this chapter is organized as follows. In next section, we provide some background knowledge on P2P systems and clustering techniques. The details of representation models and communication models are presented in Section 3. We discuss future trend and draw the conclusion in Section 4 and Section 5, respectively.

BACKGROUND

P2P Systems

Different from traditional client-server computing model, P2P systems have no central control. Each participant (peer) has equal functionality in P2P systems. Peers are autonomous and can join and leave the system at any time, which makes the systems highly dynamic. In addition, the number of peers in P2P sys-

tems is normally very large (in the range of thousands or even millions).

P2P systems display the following two nice features. First, they do not have performance bottlenecks and single points of failure. Second, P2P systems incur low deployment cost and have excellent scalability. Therefore, P2P systems have become a popular media for sharing voluminous amount of information among millions of users.

Current works in P2P systems have been focusing on efficient search. As a result, various proposals have emerged. Depending on whether some structures are enforced in the systems, existing proposals can be classified into two groups: *unstructured overlays* and *structured overlays*.

Unstructured Overlays

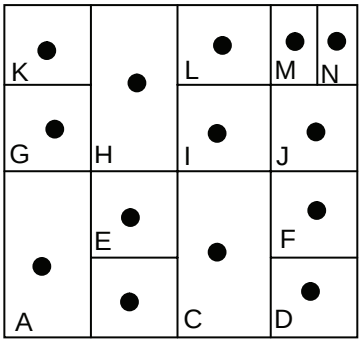
In unstructured overlays, a peer does not maintain any information about data objects stored at other peers, e.g., Gnutella. To search for a specific data object in unstructured overlays, the search message is flooded (with some constrains) to other peers in the system. While unstructured overlays are simple, they are not efficient in terms of search.

Structured Overlays

In structured overlays, e.g., CAN (Ratnasamy, 2001), CHORD (Stoica, 2001), SSW (Li, 2004), peers collaboratively maintain a distributed index structure, recording the location information of data objects shared in the system. Besides maintaining location information for some data objects, a peer also maintains a routing table with pointers pointing to a subset of peers in the system following some topology constraints. In the following, we give more details on one representative structured overlay, content addressable network (CAN).

Content Addressable Network (CAN): CAN organizes the logical data space as a *k*-dimensional Cartesian space and partitions the space into *zones*, each of which is taken charge of by a peer, called as *zone owner*. Data objects are mapped as points in the *k*-dimensional space, and the index of a data object is stored at the peer whose zone covers the corresponding point. In addition to indexing data objects, peers maintain routing tables, which consist of pointers pointing to neighboring subspaces along each dimension. Figure 1 shows one example of CAN, where data objects and peers are mapped to a 2-dimensional Cartesian space. The space is partitioned to 14 zones, and each has one peer as the zone owner.

Figure 1. Illustrative example of CAN



Clustering Algorithms

In the following, we first give a brief overview on existing clustering algorithms that are designed for centralized systems. We then provide more details on one representative density-based clustering algorithm, i.e., DBSCAN (Ester, 1996), since it is well studied in distributed environments. Nevertheless, the issues and solutions to be discussed are expected to be applicable to other clustering algorithms as well.

Overview

The existing clustering algorithms proposed for centralized systems can be classified into five classes: partition-based clustering, hierarchical clustering, grid-based clustering, density-based clustering, and model-based clustering. In the following, we provide a brief overview on these algorithms. Partition-based clustering algorithms (e.g., k-mean, MacQueen, 1967) partition *n* data objects into *k* partitions, which optimize some predefined objective function (e.g., sum of Euclidean distances to centroids). These algorithms iteratively reassign data objects to partitions and terminate when the objective function can not be improved further. Hierarchical clustering algorithms (Duda, 1973; Zhang, 1996) create a hierarchical decomposition of the data set, represented by a tree structure called dendrogram. Grid-based clustering algorithms (Agrawal, 1998; Sheikholeslami, 1998) divide the data

space into rectangular grid cells and then conduct some statistic analysis on these grid cells. Density-based clustering algorithms (Agrawal, 1998; Ankerst, 1999; Ester, 1996) cluster data objects in densely populated regions by expanding a cluster towards neighborhood with high density recursively. Model-based clustering algorithms (Cheeseman, 1996) fit the data objects to some mathematical models.

DBSCAN

The key idea of DBSCAN algorithm is that if the *r-neighborhood* of a data object (defined as the set of data objects that are within distance *r*) has a cardinality exceeding a preset threshold value *T*, this data object belongs to a cluster. In the following, we list some definition of terminologies for convenience of presentation.

Definition 1: (directly density-reachable) (Ester, 1996) An object *p* is directly density-reachable from an object *q* wrt. *r* and *T* in the set of objects *D* if

1. $p \in N_r(q)$ (where $N_r(q)$ is the subset of *D* contained in the *r*-neighborhood of *q*).
2. $N_r(q) \geq T$

Objects satisfying Property 2 in the above definition are called *core objects*. This property is called *core object property* accordingly. The objects in a cluster that are not core objects are called *border objects*. Different from noise objects that are not density-reachable from any other objects, border objects are density-reachable from some core object in the cluster.

The DBSCAN algorithm efficiently discovers clusters and noise in a dataset. According to Ester (1996), a cluster is uniquely determined by any of its core objects (Lemma 2 in Ester, 1996). Based on this fact, the DBSCAN algorithm starts from any arbitrary object *p* in the database *D* and obtains all data objects in *D* that are density-reachable from *p* through successive region queries. If *p* is a core object, the set of data objects obtained through successive region queries form a cluster in *D* containing *p*. On the other hand, if *p* is either a border object or noise object, the obtained set is empty and *p* is assigned to the noise. The above procedure is then invoked with next object in *D* that has not been examined before. This process continues till all data objects are examined.

MAIN FOCUS

In the following, we first discuss the challenging issues raised by clustering on P2P systems in Section 3.1. The solutions to these issues are then presented in Section 3.2 and 3.3.

Challenges for Clustering in P2P Systems

Clustering in P2P systems essentially consists of two steps:

- **Local clustering:** A peer (*i*) conducts clustering on its local data objects n_i.
- **Cluster assembly:** Peers collaboratively combine local clustering results to form a global clustering model.

After local clustering, some data objects form into *local clusters* while others become *local noise objects*. A local cluster is a cluster consisting of data objects that reside on one single peer. The local noise objects are data objects that are not included in any local cluster. Accordingly, we call the clusters obtained after cluster assembly as *global clusters*. A global cluster consists of one or more local clusters and/or some local noise objects.

While local clustering can adopt any existing clustering algorithm developed for centralized systems, cluster assembly is nontrivial in distributed environments. Cluster assembly algorithms in P2P systems need to overcome the following challenges:

1. **Communication efficient:** The large scale of the P2P systems and the vast amount of the data mandate the algorithm to be communication scalable in terms of the network size and data volume.
2. **Robust:** Peers might join/leave the systems or even fail at any time. Therefore, the algorithm should be sufficiently robust to accommodate these dynamic changes.
3. **Privacy Preserving:** The participants of P2P systems might come from different organizations, and they have their own interests and privacy concerns. Therefore, they would like to collaborate to obtain a global cluster result without exposing too much information to other peers from different organizations.

4. **Incremental:** The vast amount of information available in P2P systems and the dynamic change of this information repository mandate the clustering algorithm to be incremental. Any algorithms that require clustering from scratch would not be applicable in this dynamic environment.

All the above challenges concern the following two major issues:

- **Representation model:** What should be communicated among peers during cluster assembly, i.e., how the local clustering results and/or other information necessary for cluster assembly to be represented for communication among peers.
- **Communication model:** How peers communicate with each other during cluster assembly.

For completeness, in addition to reviewing solutions that are proposed for P2P systems, we also review solutions that are proposed for distributed systems. The difference between distributed systems and P2P systems is that distributed systems have much smaller scale, and a centralized server is normally present to perform certain coordination in distributed systems. If necessary, we explicitly point out the deficiency of the solutions proposed for distributed systems.

Representation Model

Ideally, a representation model should satisfy the following two criterions:

- The size of the representation model should be small.
- The model should accurately represent local clustering results and/or necessary information for cluster assembly.

In this section, we review three representation models, i.e., two approximate models based on core objects (Januzaj, 2004) and one exact model based on spatial boundary (Li, 2006). While core object based models tradeoff accuracy for compactness through approximation, spatial boundary based model leverages spatial properties to compactly represent necessary information for cluster assembly.

Approximate Representation Model: Core Object based Model

Januzaj (2004) proposed two models, i.e., Rep_{scor} and $\text{Rep}_{\text{k-means}}$, which approximate local clustering results using some representatives. Both of these two models are based on the concept of *specific core objects*. A set of specific core objects is defined as the set of core objects of a local cluster satisfying the following two conditions:

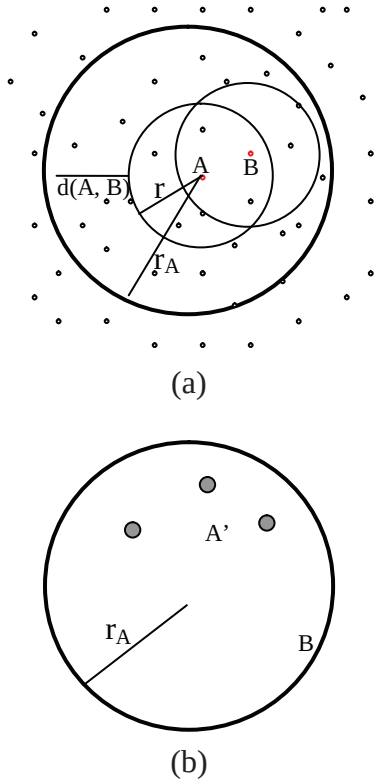
- Each data object in the local cluster is directly density reachable from one of the specific core objects.
- Any pair of specific objects is not directly density reachable from each other.

The set of specific core objects might not be unique, and which data objects are included in the set depends on the visiting order of data objects during clustering. For instance, in Figure 2(a), both *A* and *B* are core objects, and they are in each other's *r*-neighborhood. Assuming *A* is examined first during clustering process and it is included in the set of specific core objects, *B* will not be included in the set.

Rep_{scor}. Rep_{scor} represents each local cluster by a set of specific core data objects, which not only represent the data objects in its *r*-neighborhood, but also represent the data objects in the r-neightbood of the core data obects within its r-neighborhood.

$\text{Rep}_{\text{k-means}}$ Rep_{scor}can not always represent the data objects in a local cluster accurately. As shown in Figure 2(b), *A* is a specific core object. The combination of *A* and its *r*-range (A, r_A) represent the data objects in the *e*-neighborhood of *A* (i.e., *A* itself and the other 4 data objects depicted by grey circles). However, the five data objects are not uniformly distributed in the neighborhood. This motivates the proposal of another representation model, $\text{Rep}_{\text{k-means}}$, which represents a local cluster using the centroids obtained by k-mean clustering over the data objects of this local cluster. Figure 2(b) shows one example for $\text{Rep}_{\text{k-means}}$. After k-mean clustering, *A'* is obtained as the centroid to represent the data objects in the *r*-neighborhood of *A*. It is obvious that *A'* can better represent these data objects than *A* can.

Figure 2. An illustrative example of Rep_{scor} and $Rep_{k\text{-}means}$



Exact Representation Model: Spatial Boundary Based Model

When the peers and data objects are organized systematically as in structured overlays, not all local clustering results are necessary for cluster assembly. Instead, only the data objects residing along the zone boundary might affect clustering. This motivates Li (2006) to propose a spatial boundary based model, which represents the information necessary for cluster assembly in a compact format by leveraging the structures among peers and data objects in structured overlays.

We define the region inside Zone *i* that is within *r* to the boundary of Zone *i* as the *r-inner-boundary* of Zone *i*, and the region outside of Zone *i* that is within *r* to the boundary of Zone *i* as the *r-outer-boundary* of Zone *i*. Li (2006) has the following two observations about structured overlays, or more specifically CAN overlays:

- If a local cluster (within a zone) has no data object residing in the *r*-inner-boundary of the zone, this local cluster is *non-expandable*, i.e., it is a global cluster.
- If a local cluster within a zone can be expanded to other zones (i.e., this local cluster is *expandable*), there must exist some data objects belong to this local cluster in the *r*-inner-boundary of this zone, and the *r*-neighborhood of such data objects contains some data objects in the *r*-outer-boundary of this zone.

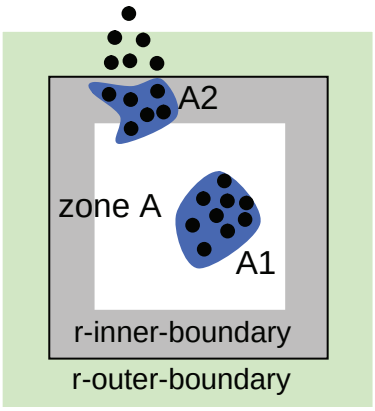
Figure 3 illustrates examples of an expandable cluster (*A2*) and a non-expendable cluster (*A1*) within Zone *A* (depicted by the rectangle).

Based on these observations, spatial boundary based model is represented by *cluster expansion set (CES)*, which consists of *present coverage (Pcoverage)* and *expandable coverage (Ecoverage)*, where Pcoverage includes the LocalClusterID (a tuple of <zoneID, clusterID>) for a local expandable cluster within the zone, and Ecoverage includes the zoneID and LocalClusterIDs for the local clusters or noise objects in the *r*-outer-boundary that this local cluster can be expanded to.

Communication Model

For completeness, we review three possible communication models, namely flooding-based communication model, centralized communication model, and hierarchical communication model. While the former two are proposed for distributed systems, the latter one is proposed for P2P systems.

Figure 3. Expandable and non-expandable clusters.



Flooding-Based Communication Model

In flooding-based communication model (e.g., Dhillon (1999) and Forman (2000)), each peer floods its local clustering result to all other peers. Eventually every peer has a global view of local clustering results on all peers and can form global clustering result individually. The disadvantage of flooding-based communication model is the excessive communication overheads incurred by multiple rounds of message flooding.

Centralized Communication Model

In centralized communication model (e.g., Januzaj (2004), Johnson (1999), Samatova (2002) and Xu (1999)), each peer forwards its local clustering result to a designated central site. This central site combines the local clustering results to form global clustering result. Although centralized communication model is simple, it requires a designated server.

Hierarchical Communication Model

Motivated by the above deficiencies of flooding-based communication model and centralized communication model, Li (2006) proposed a hierarchical communication model for cluster assembly. This hierarchical communication model does not require a central site, nor does it require system-wide flooding. By leveraging CAN overlay (where data objects are mapped to zones), this hierarchical communication model enables peers to assemble local clusters within smaller regions, which are then recursively assembled to form global clusters. The proposed hierarchical communication model does not impose high processing load at the root or nodes at the high levels of the hierarchy.

FUTURE TRENDS

The new challenges and opportunities raised by data mining over the newly emerged networked systems will attract a rich body of interdisciplinary studies from data mining and networking communities. While various data mining tasks will have specific issues of their own to address, performing the localized data mining and assembly of local mining results in an integrated manner is an important research issue to address and certainly needs more research effort from both research communities. We anticipate a growing number of studies on developing new techniques for addressing these issues.

CONCLUSION

The emergence of networked data sources (e.g., P2P systems, sensor networks) brings new challenges and opportunities for data mining. The large scale and high dynamics of P2P systems mandate the data mining algorithms designed for such systems to be communication efficient, robust, privacy preserving, and incremental.

In this chapter, we investigate clustering issue, one of the important data mining tasks, in P2P systems. We identify two important issues, i.e., representation model and communication model, raised by clustering in P2P systems, and review the start-of-the-art solutions in the literature to address these two issues.

ACKNOWLEDGMENT

This work was supported in part by the National Science Foundation under Grant no. IIS-0534343, IIS-0328881 and CNS-0626709.

REFERENCES

R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan (1998). Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of SIGMOD*, pages 94–105.

M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander (1999). OPTICS: Ordering points to identify the clustering structure. In *Proceedings of SIGMOD*, pages 49–60.

P. Cheeseman and J. Stutz (1996). Bayesian classification (autoclass): Theory and results. In *Advances in Knowledge Discovery and Data Mining*, pages 153–180. AAAI/MIT Press.

I. S. Dhillon and D. S. Modha. A data-clustering algorithm on distributed memory multiprocessors (1999). In *Proceedings of Workshop on Large-Scale Parallel KDD Systems (in conjunction with SIGKDD)*, pages 245–260.

R. Duda and P. Hart. *Pattern classification and scene analysis*. John Wiley & Sons, New York, 1973.

M. Ester, H.-P. Kriegel, J. Sander, and X. Xu (1996). A density based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of Knowledge Discovery in Database (KDD)*, pages 226–231.

G. Forman and B. Zhang (2000). Distributed data clustering can be efficient and exact. *SIGKDD Explorations*, 2(2):34–38.

E. Januzaj, H.-P. Kriegel, and M. Pfeifle (2004). DBDC: Density based distributed clustering. In *Proceedings of International Conference on Extending Database Technology (EDBT)*, pages 88–105.

E. L. Johnson and H. Kargupta (1999). Collective, hierarchical clustering from distributed, heterogeneous data. In *Proceedings of Workshop on Large-Scale Parallel KDD Systems (in conjunction with SIGKDD)*, pages 221–244.

M. Li, W.-C. Lee, and A. Sivasubramaniam (2004). Semantic small world: An overlay network for peer-to-peer search. In *Proceedings of International Conference on Network Protocols (ICNP)*, pages 228–238.

M. Li, G. Lee, W.-C. Lee and and A. Sivasubramaniam (2006). PENS: An algorithm for density-based clustering in peer-to-peer systems. In *Proceedings of International Conference on Scalable Information Systems (INFOSCALE)*.

J. MacQueen (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297.

S. Ratnasamy, P. Francis, M. Handley, R. M. Karp, and S. Schenker (2001). A scalable content-addressable network. In *Proceedings of ACM SIGCOMM*, pages 161–172.

N. F. Samatova, G. Ostrouchov, A. Geist, and A. V. Melechko (2002). RACHET: An efficient cover-based merging of clustering hierarchies from distributed datasets. *Distributed and Parallel Databases*, 11(2):157–180.

G. Sheikholeslami, S. Chatterjee, and A. Zhang (1998). WaveCluster: A multi-resolution clustering approach for very large spatial databases. In *Proceedings of VLDB*, pages 428–439.

I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan (2001). Chord: A scalable peer-to-peer lookup service for Internet applications. In *Proceedings of ACM SIGCOMM*, pages 149–160.

X. Xu, J. Jäger, and H.-P. Kriegel (1999). A fast parallel clustering algorithm for large spatial databases. *Data Mining and Knowledge Discovery*, 3(3):263–290.

T. Zhang, R. Ramakrishnan, and M. Livny (1996). BIRCH: An efficient data clustering method for very large databases. In *Proceedings of SIGMOD*, pages 103–114.

KEY TERMS

Cluster Assembly: The process of assembling local cluster results into global cluster results.

Clustering: One of the most important data mining tasks. Clustering is to group a set of data objects into classes consisting of similar data objects.

Data Mining: The process of extracting previous unknown and potentially valuable knowledge from a large set of data.

Density-based Clustering: An representative clustering technique that treats densely populated regions separated by sparsely populated regions as clusters.

Overlay Networks: A computer network built on top of other networks, where the overlay nodes are connected by logical links corresponding to several physical links in the underlying physical networks. Peer-to-peer systems are one example of overlay networks.

Peer-To-Peer Systems: A system consisting of a large number of computer nodes, which have relatively equal responsibility.

P2P Data Mining: The process of data mining in peer-to-peer systems.

Clustering of Time Series Data

Anne Denton
North Dakota State University, USA

INTRODUCTION

Time series data is of interest to most science and engineering disciplines and analysis techniques have been developed for hundreds of years. There have, however, in recent years been new developments in data mining techniques, such as frequent pattern mining, that take a different perspective of data. Traditional techniques were not meant for such pattern-oriented approaches. There is, as a result, a significant need for research that extends traditional time-series analysis, in particular clustering, to the requirements of the new data mining algorithms.

BACKGROUND

Time series clustering is an important component in the application of data mining techniques to time series data (Roddick, Spiliopoulou, 2002) and is founded on the following research areas:

- **Data Mining:** Besides the traditional topics of classification and clustering, data mining addresses new goals, such as frequent pattern mining, association rule mining, outlier analysis, and data exploration (Tan, Steinbach, and Kumar 2006).
- **Time Series Data:** Traditional goals include forecasting, trend analysis, pattern recognition, filter design, compression, Fourier analysis, and chaotic time series analysis. More recently frequent pattern techniques, indexing, clustering, classification, and outlier analysis have gained in importance.
- **Clustering:** Data partitioning techniques such as k-means have the goal of identifying objects that are representative of the entire data set. Density-based clustering techniques rather focus on a description of clusters, and some algorithms identify the most common object. Hierarchical techniques define clusters at multiple levels of granularity. A survey of clustering that includes its application to time series data is provided in (Gan, Ma, and Wu, 2007).
- **Data Streams:** Many applications, such as communication networks, produce a stream of data (Muthukrishnan, 2003). For real-valued attributes such a stream is amenable to time series data mining techniques.

Time series clustering draws from all of these areas. It builds on a wide range of clustering techniques that have been developed for other data, and adapts them while critically assessing their limitations in the time series setting.

MAIN THRUST OF THE CHAPTER

Many specialized tasks have been defined on time series data. This chapter addresses one of the most universal data mining tasks, clustering, and highlights the special aspects of applying clustering to time series data. Clustering techniques overlap with frequent pattern mining techniques, since both try to identify typical representatives.

Clustering Time Series

Clustering of any kind of data requires the definition of a similarity or distance measure. A time series of length n can be viewed as a vector in an n-dimensional vector space. One of the best-known distance measures, Euclidean distance, is frequently used in time series clustering. The Euclidean distance measure is a special case of an L_p norm. L_p norms may fail to capture similarity well when being applied to raw time series data because differences in the average value and average derivative affect the total distance. The problem is typically addressed by subtracting the mean and dividing the resulting vector by its L_2 norm, or by working with normalized derivatives of the data

(Gavrilov et al., 2000). Several specialized distance measures have been used for time series clustering, such as dynamic time warping, DTW (Berndt and Clifford 1996), longest common subsequence similarity, LCSS (Vlachos, Gunopulos, and Kollios, 2002), and a distance measure based on well-separated geometric sets (Bollabas, Das, Gunopulos, and Mannila 1997).

Some special time series are of interest. A strict white noise time series is a real-valued sequence with values $X_t = e_t$ where e_t is Gaussian distributed random variable. A random walk time series satisfies $X_t - X_{t-1} = e_t$ where e_t is defined as before.

Time series clustering can be performed on whole sequences or on subsequences. For clustering of whole sequences, high dimensionality is often a problem. Dimensionality reduction may be achieved through Discrete Fourier Transform (DFT), Discrete Wavelet Transform (DWT), and Principal Component Analysis (PCA), as some of the most commonly used techniques. DFT (Agrawal, Falutsos, and Swami, 1993) and DWT have the goal of eliminating high-frequency components that are typically due to noise. Specialized models have been introduced that ignore some information in a targeted way (Jin, Lu, and Shi 2002). Others are based on models for specific data such as socioeconomic data (Kalpakis, Gada, and Puttagunta, 2001).

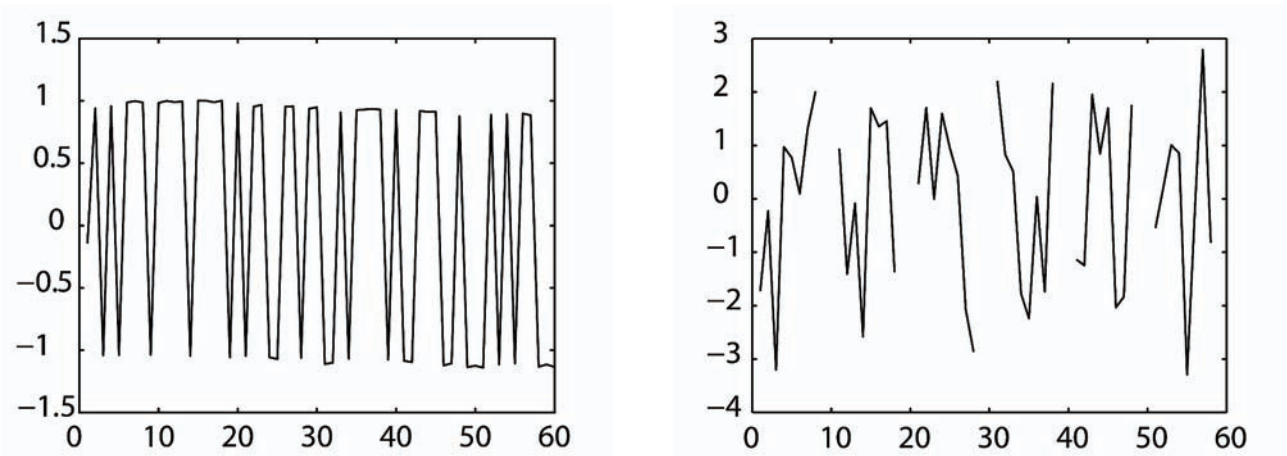
A large number of clustering techniques have been developed, and for a variety of purposes (Halkidi, Batistakis, and Vazirgiannis, 2001). Partition-based techniques are among the most commonly used ones for time series data. The k-means algorithm, which is based on a greedy search, has recently been generalized to a wide range of distance measures (Banerjee et al., 2004).

Clustering Subsequences of a Time Series

A variety of data mining tasks require clustering of subsequences of one or more time series as preprocessing step, such as Association Rule Mining (Das et al., 1998), outlier analysis and classification. Partition-based clustering techniques have been used for this purpose in analogy to vector quantization (Gersho and Gray, 1992) that has been developed for signal compression. It has, however, been shown that when a large number of subsequences are clustered, the resulting cluster centers are very similar for different time series (Keogh, Lin, and Truppel, 2003). Figure 1 illustrates how k-means clustering may find cluster centers that do not represent any part of the actual time series. Note how the short sequences in the right panel (cluster centers) show patterns that do not occur in the time series (left panel) which only has two possible values. A mathematical derivation why cluster centers in k-means are expected to be sinusoidal in certain limits has been provided for k-means clustering (Ide, 2006).

Several solutions have been proposed, which address different properties of time series subsequence data, including trivial matches that were observed in

Figure 1. Time series glassfurnace (left) and six cluster centers that result from k-means clustering with k=6 and window size w=8.



(Keogh, Lin, and Truppel, 2003), the relationship to dynamical systems (Chen, 2005), and the frequently occurring similarity to random walk noise (Denton, 2005). One approach is based on adapting density-based clustering techniques (Hinneburg and Keim, 2003) which are robust to noise, to time series data (Denton, 2004) and (Denton, 2005). Partition-based techniques aim at finding representatives for all objects. Cluster centers in kernel-density-based clustering, in contrast, are sequences that are in the vicinity of the largest number of similar objects. The goal of kernel-density-based techniques is thereby similar to frequent-pattern mining techniques such as Motif-finding algorithms (Patel et al., 2002). The concept of cluster cores (Peker, 2005) also captures the idea of identifying representative sequences, and the work furthermore suggests that using a large number of clusters can alleviate problems in time series subsequence data, in domains where that is an acceptable result. Other solutions exist, such as ones based on techniques that were developed for dynamical systems (Chen, 2005). These techniques work well for time series, in which patterns occur at regular intervals. Pattern mining concepts have also been successfully applied to time series subsequence clustering (Fu, Chung, Luk, and Ng, 2005) and an unfolding preprocessing method has been developed (Simon, Lee, and Verleysen, 2006). Finally the problem has been discussed from the perspective of distinguishing entire clusterings (Goldin, Mardales, and Nagy, 2006).

Related Problems

One time series clustering problem of particular practical relevance is clustering of gene expression data (Eisen, Spellman, Brown, and Botstein, 1998). Gene expression is typically measured at several points in time (time course experiment) that may not be equally spaced. Hierarchical clustering techniques are commonly used. Density-based clustering has recently been applied to this problem (Jiang, Pei, and Zhang, 2003)

Time series with categorical data constitute a further related topic. Examples are log files and sequences of operating system commands. Some clustering algorithms in this setting borrow from frequent sequence mining algorithms (Vaarandi, 2003).

Mining of data streams is also closely related. Common applications are computer network traffic, sensor networks, and web logs (Gaber, Zaslavsky, and Krishnaswamy, 2005).

FUTURE TRENDS

Storage grows exponentially at rates faster than Moore's law for microprocessors. A natural consequence is that old data will be kept when new data arrives leading to a massive increase in the availability of the time-dependent data. Data mining techniques will increasingly have to consider the time dimension as an integral part of other techniques. Much of the current effort in the data mining community is directed at data that has a more complex structure than the simple tabular format initially covered in machine learning (Džeroski and Lavrač, 2001). Examples, besides time series data, include data in relational form, such as graph- and tree-structured data, and sequences. A major step towards integrating diverse data into a combined data mining framework consists in finding symbolic representations for different types of data. Time series subsequence clustering constitutes one such approach towards defining a mapping to a symbolic representation (Das et al., 1998). When addressing new settings it will be of major importance to not only generalize existing techniques and make them more broadly applicable but to also critically assess problems that may appear in the generalization process.

CONCLUSION

Despite the maturity of both clustering and time series analysis, time series clustering is an active and fascinating research topic. New data mining applications are constantly being developed and require new types of clustering results. Clustering techniques from different areas of data mining have to be adapted to the time series context. Noise is a particularly serious problem for time series data, thereby adding challenges to clustering process. Considering the general importance of time series data, it can be expected that time series clustering will remain an active topic for years to come.

REFERENCES

Banerjee, A., Merugu, S., Dhillon, I., & Ghosh, J. (2004, April). Clustering with Bregman divergences. In Proceedings *SIAM International Conference on Data Mining*, Lake Buena Vista, FL.

Berndt D.J., Clifford, J. (1996). Finding patterns in time series: a dynamic programming approach. *Advances in knowledge discovery and data mining*, AAAI Press, Menlo Park, CA, 229 – 248.

Bollobas, B., Das, G., Gunopulos, D., Mannila, H. (1997). Time-series similarity problems and well-separated geometric sets. In Proceedings *13th annual symposium on Computational geometry*, Nice, France, 454 - 456.

Chen, J.R. (2005). Making time series clustering meaningful. In Proceedings *5th IEEE International Conference on Data Mining*, Houston, TX, 114-121.

Das, G., Gunopulos, D. (2003). Time series similarity and indexing. In *The Handbook of Data Mining*, Ed. Ye, N., Lawrence Erlbaum Associates, Mahwah, NJ, 279-304.

Das, G., Lin, K.-I., Mannila, H., Renganathan, G., & Smyth, P. (1998, Sept). Rule discovery from time series. In Proceedings *IEEE Int. Conf. on Data Mining*, Rio de Janeiro, Brazil.

Denton, A. (2005, Aug.). Kernel-density-based clustering of time series subsequences using a continuous random-walk noise model. In Proceedings 5th IEEE International Conference on Data Mining, Houston, TX, 122-129.

Denton, A. (2004, Aug.). Density-based clustering of time series subsequences. In Proceedings *The Third Workshop on Mining Temporal and Sequential Data (TDM 04) in conjunction with The Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Seattle, WA.

Džeroski & Lavrač (2001). *Relational Data Mining*. Berlin, Germany: Springer.

Eisen, M.B., Spellman, P.T., Brown, P.O., Botstein, D. (1998, Dec.). Cluster analysis and display of genome-wide expression patterns. In Proceedings *Natl. Acad. Sci. U.S.A.*; 95(25), 14863-8.

Fu, T., Chung, F, Luk, R., and Ng, C. (2005), Preventing meaningless stock time series pattern discovery by change perceptually important point detection, in Proceedings *2nd International Conference on Fuzzy Systems and Knowledge Discovery*, Changsha, China, 1171-1174.

Gaber, M.M., Krishnaswamy, S., and Zaslavsky, A.B. (2005) Mining data streams: a review. *SIGMOD Record 34(2)*,18-26.

Gan, G., Ma, C., and Wu, J. (2007). *Data Clustering: Theory, Algorithms, and Applications,*. SIAM, Society for Industrial and Applied Mathematics.

Gavrilov, M., Anguelov, D., Indyk, P., Motwani, R. (2000). Mining the stock market (extended abstract): which measure is best? In Proceedings *Sixth ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining* Boston, MA, 487 – 496.

Gersho, A., Gray, R.M. (1992) *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, Boston, MA.

Goldin, D., Mardales, R., & Nagy, G. (2006 Nov.). In search of meaning for time series subsequence clustering: Matching algorithms based on a new distance measure. In Proceedings Conference on Information and Knowledge Management, Washington, DC.

Halkidi, M., Batistakis, & Y., Vazirgiannis, M. (2001). On clustering validation techniques, *Intelligent Information Systems Journal*, Kluwer Pulishers, 17(2-3), 107-145.

Hinneburg, A. & Keim, D.A. (2003, Nov.). A general approach to clustering in large databases with noise. *Knowl. Inf. Syst* 5(4), 387-415.

Ide, T. (2006 Sept.). Why does subsequence time series clustering produce sine waves? In Proceedings *10th European Conference on Principles and Practice of Knowledge Discovery in Databases, Lecture Notes in Artificial Intelligence* 4213, Springer, 311-322.

Jiang, D., Pei, J., & Zhang, A. (2003, March). DHC: A Density-based hierarchical clustering method for time series gene expression data. In Proceedings *3rd IEEE Symp. on Bioinformatics and Bioengineering (BIBE'03)*, Washington D.C.

Jin, X., Lu, Y., Shi, C. (2002). Similarity measure based on partial information of time series. In Proceedings *Eighth ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, Edmonton, AB, Canada, 544 – 549.

Kalpakis, K., Gada, D., & Puttagunta, V. (2001). *Distance measures for effective clustering of ARIMA*

time-series. In Proceedings *IEEE Int. Conf. on Data Mining*. San Jose, CA, 273-280.

Keogh, E.J., Lin, J., & Truppel, W. (2003, Dec). Clustering of time series subsequences is meaningless: implications for previous and future research. In Proceedings *IEEE Int. Conf. on Data Mining*, Melbourne, FL, 115-122

Muthukrishnan, S. (2003). Data streams: algorithms and applications. In: Proceedings *Fourth annual ACM-SIAM symposium on discrete algorithms.* Baltimore, MD,

Patel, P., Keogh, E., Lin, J., & Lonardi, S. (2002). Mining motifs in massive time series databases. In Proceedings *2002 IEEE Int. Conf. on Data Mining*. Maebashi City, Japan.

Peker, K. (2005). Subsequence time series clustering techniques for meaningful pattern discovery. In Proceedings IEE KIMAS Conference.

Reif, F. (1965). *Fundamentals of Statistical and Thermal Physics*, New York, NY: McGraw-Hill.

Roddick, J.F. & Spiliopoulou, M. (2002). A survey of temporal knowledge discovery paradigms and methods. *IEEE Transactions on Knowledge and Data Engineering* **14**(4), 750-767.

Simon, G., Lee, J.A., & Verleysen, M. (2006). Unfolding preprocessing for meaningful time series clustering. Neural Networks, 19(6), 877-888.

Tan, P.-N., Steinbach, M. & Kumar, V. (2006). *Introduction to Data Mining*. Addison-Wesley.

Vlachos, M., Gunopoulos, D., & Kollios, G., (2002, Feb.). Discovering Similar Multidimensional Trajectories. In Proceedings *18th International Conference on Data Engineering* (ICDE'02), San Jose, CA.

Vaarandi, R. (2003). A data clustering algorithm for mining patterns from event logs. In Proceedings *2003 IEEE Workshop on IP Operations and Management*, Kansas City, MO.

KEY TERMS

Dynamic Time Warping (DTW): Sequences are allowed to be extended by repeating individual time series elements, such as replacing the sequence X={x1,x2,x3} by X'={x1,x2,x2,x3}. The distance between two sequences under dynamic time warping is the minimum distance that can be achieved in by extending both sequences independently.

Kernel-Density Estimation (KDE): Consider the vector space in which the data points are embedded. The influence of each data point is modeled through a kernel function. The total density is calculated as the sum of kernel functions for each data point.

Longest Common Subsequence Similarity (LCSS): Sequences are compared based on the assumption that elements may be dropped. For example, a sequence X={x1,x2,x3} may be replaced by X"={x1,x3}. Similarity between two time series is calculated as the maximum number of matching time series elements that can be achieved if elements are dropped independently from both sequences. Matches in real-valued data are defined as lying within some predefined tolerance.

Partition-Based Clustering: The data set is partitioned into k clusters, and cluster centers are defined based on the elements of each cluster. An objective function is defined that measures the quality of clustering based on the distance of all data points to the center of the cluster to which they belong. The objective function is minimized.

Principle Component Analysis (PCA): The projection of the data set to a hyper plane that preserves the maximum amount of variation. Mathematically PCA is equivalent to singular value decomposition on the covariance matrix of the data.

Random Walk: A sequence of random steps in an n-dimensional space, where each step is of fixed or randomly chosen length. In a random walk time series, time is advanced for each step and the time series element is derived using the prescription of a 1-dimensional random walk of randomly chosen step length.

Sliding Window: A time series of length n has $(n-w+1)$ subsequences of length w. An algorithm that operates on all subsequences sequentially is referred to as a sliding window algorithm.

Time Series: Sequence of real numbers, collected at equally spaced points in time. Each number corresponds to the value of an observed quantity.

Vector Quantization: A signal compression technique in which an n-dimensional space is mapped to a finite set of vectors. Each vector is called a codeword and the collection of all codewords a codebook. The codebook is typically designed using Linde-Buzo-Gray (LBG) quantization, which is very similar to k-means clustering.

On Clustering Techniques

Sheng Ma
Machine Learning for Systems IBM T. J. Watson Research Center, USA

Tao Li
School of Computer Science, Florida International University, USA

INTRODUCTION

Clustering data into sensible groupings, as a fundamental and effective tool for efficient data organization, summarization, understanding and learning, has been the subject of active research in several fields such as statistics (Jain & Dubes, 1988; Hartigan, 1975), machine learning (Dempster, Laird & Rubin, 1977), Information theory (Linde, Buzo & Gray, 1980), databases (Guha, Rastogi & Shim, 1998; Zhang, Ramakrishnan & Livny, 1996) and Bioinformatics (Cheng & Church, 2000) from various perspectives and with various approaches and focuses. From application perspective, clustering techniques have been employed in a wide variety of applications such as customer segregation, hierarchal document organization, image segmentation, microarray data analysis and psychology experiments.

Intuitively, the clustering problem can be described as follows: Let $\boldsymbol{W}$ be a set of n entities, finding a partition of $\boldsymbol{W}$ into groups such that the entities within each group are **similar** to each other while entities belonging to different groups are **dissimilar**. The entities are usually described by a set of measurements (attributes). Clustering does not use category information that labels the objects with prior identifiers. The absence of label information distinguishes cluster analysis from classification and indicates that the goals of clustering is just finding a hidden structure or compact representation of data instead of discriminating future data into categories.

BACKGROUND

Generally clustering problems are determined by five basic components:

- **Data representation:** What's the (physical) representation of the given data set? What kind of attributes (e.g., numerical, categorical or ordinal)?
- **Data generation:** The formal model for describing the generation of the data set. For example, Gaussian mixture model is a model for data generation.
- **Criterion/objective function:** What are the objective functions or criteria that the clustering solutions should aim to optimize? Typical examples include entropy, maximum likelihood and within-class or between-class distance (Li, Ma & Ogihara, 2004a).
- **Optimization procedure:** What is the optimization procedure for finding the solutions? Clustering problem is known to be NP-complete (Brucker, 1977) and many approximation procedures have been developed. For instance, Expectation-Maximization (EM) type algorithms have been widely used to find local minima of optimization.
- **Cluster validation and interpretation**: Cluster validation evaluates the clustering results and judges the cluster structures. Interpretation is often necessary for applications. Since there is no label information, clusters are sometimes justifies by ad hoc methods (such as exploratory analysis) based on specific application areas.

For a given clustering problem, the five components are tightly coupled. The formal model is induced from the physical representation of the data, the formal model along with the objective function determines the clustering capability and the optimization procedure decides how efficiently and effectively the clustering results can be obtained. The choice of the optimization procedure depends on the first three components. Validation of cluster structures is a way of verifying assumptions on data generation and evaluating the optimization procedure.

MAIN THRUST

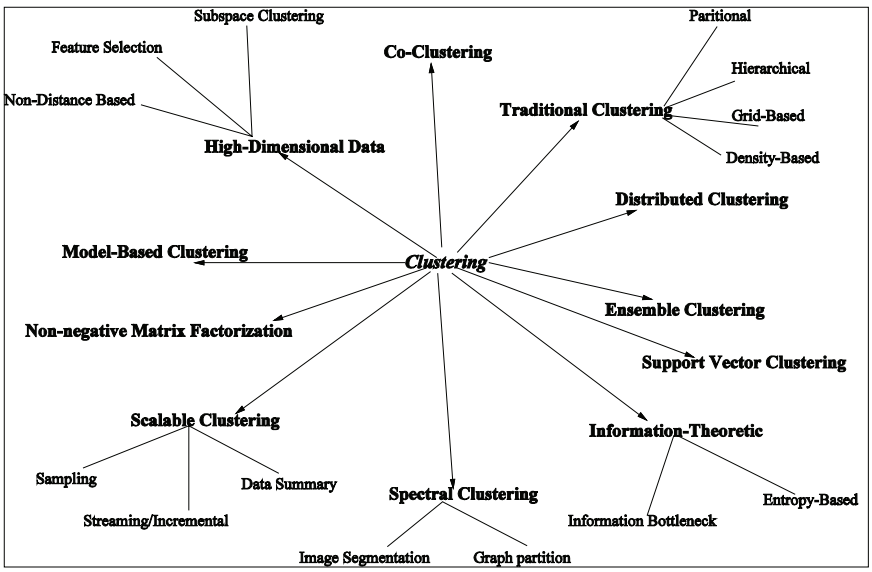
We review some of current clustering techniques in the section. Figure 1 gives the summary of clustering techniques. The following further discusses traditional clustering techniques, spectral based analysis, model-based clustering and co-clustering.

Traditional clustering techniques focus on one-sided clustering and they can be classified into partitional, hierarchical, density-based, and grid-based (Han & Kamber, 2000). Partitional clustering attempts to directly decompose the data set into disjoint classes such that the data points in a class are nearer to one another than the data points in other classes. Hierarchical clustering proceeds successively by building a tree of clusters. Density-based clustering is to group the neighboring points of a data set into classes based on density conditions. Grid-based clustering quantizes the data space into a finite number of cells that form a grid-structure and then performs clustering on the grid structure. Most of these algorithms use distance functions as objective criteria and are not effective in high dimensional spaces.

As an example, we take a closer look at K-means algorithms. The typical K-means type algorithm is a widely-used partitional-based clustering approach. Basically, it first chooses a set of K data points as initial cluster representatives (e.g., centers), and then performs an iterative process that alternates between assigning the data points to clusters based on their distances to the cluster representatives and updating the cluster representatives based on new cluster assignments. The iterative optimization procedure of K-means algorithm is a special form of EM-type procedure. The K-means type algorithm treats each attribute equally and computes the distances between data points and cluster representatives to determine cluster memberships.

A lot of algorithms have been developed recently to address the efficiency and performance issues presented in traditional clustering algorithms. Spectral analysis has been shown to tightly relate to clustering task. Spectral clustering (Weiss, 1999; Ng, Jordan & Weiss, 2001), closely related to the latent semantics index (LSI), uses selected eigenvectors of the data affinity matrix to obtain a data representation that can be easily clustered or embedded in a low-dimensional space. Model-based clustering attempts to learn generative models, by which the cluster structure is determined, from the data. (Tishby, Pereira & Bialek, 1999; Slonim & Tishby, 2000) develop information bottleneck formulation, in which given the empirical joint distribution of two variables, one variable is compressed so that the mutual information about the other is preserved as much as possible. Other recent developments of clustering techniques include ensemble clustering, support vector clustering, matrix factorization, high-dimensional data clustering, distributed clustering and etc.

Figure 1. Summary of clustering techniques



Another interesting development is co-clustering that conducts simultaneous, iterative clustering of both data points and their attributes (features) through utilizing the canonical duality contained in the point-by-attribute data representation. The idea of co-clustering of data points and attributes dates back to (Anderberg, 1973; Nishisato, 1980). Govaert (1985) researches simultaneous block clustering of the rows and columns of contingency table. The idea of co-clustering has been also applied to cluster gene expression and experiments (Cheng & Church, 2000). Dhillon (2001) presents a co-clustering algorithm for documents and words using bipartite graph formulation and a spectral heuristic. Recently Dhillon et al. (2003) propose an information-theoretic co-clustering method for two-dimensional contingency table. By viewing the non-negative contingency table as a joint probability distribution between two discrete random variables, the optimal co-clustering then maximizes the mutual information between the clustered random variables. Li and Ma (2004) recently develop Iterative Feature and Data clustering (IFD) by representing the data generation with data and feature coefficients. IFD enables an iterative co-clustering procedure for both data and feature assignments. However, unlike previous co-clustering approaches, IFD performs clustering using the mutually reinforcing optimization procedure that has proven convergence property. IFD only handles data with binary feature. Li, Ma and Ogihara (2004b) further extend the idea for general data.

FUTURE TRENDS

Although clustering has been studied for many years, many issues such as cluster validation still need more investigation. In addition, new challenges such as scalability, high-dimensionality and complex data types, have been brought by the ever-increasing growth of information exposure and data collection.

- **Scalability and efficiency:** With the collection of huge amount of data, clustering faces the problems of scalability in terms of both computation time and memory requirements. To resolve the scalability issues, methods such as incremental and streaming approaches, sufficient statistics for data summary and sampling techniques have been developed.
- **Curse of dimensionality**: Another challenge is the high dimensionality of data. It has been shown that in a high dimensional space, the distance between every pair of points is almost the same for a wide variety of data distributions and distance functions (Beyer et al., 1999). Hence most algorithms do not work efficiently in high dimensional spaces due to the **curse of dimensionality**. Many feature selection techniques have been applied to reduce the dimensionality of the space. However, as demonstrated in (Aggarwal et al., 1999), in many case, the correlations among the dimensions are often specific to data locality; in other words, some data points are correlated with a given set of features and others are correlated with respect to different features. As pointed out in (Hastie, Tibshirani & Friedman, 2001; Domeniconi, Gunopulos & Ma, 2004), all methods that overcome the dimensionality problems use a metric for measuring neighborhoods, which is often implicit and/or adaptive.
- **Complex data types:** The problem of clustering becomes more challenging when the data contains complex types. For example, when the attributes contain both categorical and numerical values. There are no inherent distance measures between data values. This is often the case in many applications where data are described by a set of descriptive or presence/absence attributes, many of which are not numerical. The presence of complex types also makes the cluster validation and interpretation difficult.

More challenges also include clustering with multiple criteria (where clustering problems often require optimization over more than one criterion), clustering relation data (where data is represented with multiple relation tables) and distributed clustering (where data sets are geographically distributed across multiple sites).

CONCLUSION

Clustering is a classical topic to segment and group similar data objects. Many algorithms have been developed in the past. Looking ahead, many challenges drawn from real-word applications will drive the search for efficient algorithms that are able to handle hetero-

geneous data, to process a large volume, and to scale to deal with a large number of dimensions.

REFERENCES

Aggarwal, C., Wolf, J.L., Yu, P.S., Procopiuc, C., & Park, J. S. Park (1999). Fast algorithms for projected clustering. In *Proceedings of ACM SIGMOD Conference.* 61-72.

Anderberg, M.~R. (1973). *Cluster analysis for applications.* Academic Press Inc.

Beyer, K, Goldstein, J., Ramakrishnan, R. & Shaft, U. (1999). When is nearest neighbor meaningful? In *Proceedings of International Conference on Database Theory (ICDT),* 217-235.

Brucker, P. (1977). On the complexity of clustering problems. In: R. Henn, B. Korte, and W. Oletti, editors, *Optimization and Operations Research,* Heidelberg, New York, NY, 45-54. Springer-Verlag.

Cheng, Y., & Church, G.M. (2000) Bi-clustering of expression data. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology (ISMB),* 93-103.

Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, 39*, 1-38.

Dhillon, I. (2001). *Co-clustering documents and words using bipartite spectral graph partitioning.* (Technical Report 2001-05). UT Austin CS Dept.

Dhillon, I.S., Mallela, S. & Modha, S.~S. (2003). Information-theoretic co-clustering. In *Proceedings of ACM SIGKDD 2003*, 89-98.

Domeniconi, C., Gunopulos, D., & Ma, S. (2004). Within-cluster adaptive metric for clustering. In *Proceedings of SIAM International Conference on Data Mining.*

Guha, S., Rastogi, R., & Shim, K. (1998). CURE: an efficient clustering algorithm for large databases. In *Proceedings of ACM SIGMOD Conference,* 73-84.

Govaert, G. (1985). Simultaneous clustering of rows and columns. *Control and Cybernetics*. 437-458.

Hartigan, J. (1975). *Clustering algorithms.* Wiley.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning: data mining, inference, prediction.* Springer.

Han, J. & Kamber, M. (2000). *Data mining: Concepts and techniques.* Morgan Kaufmann Publishers.

Jain, A.K. & Dubes, R.C. (1988). *Algorithms for clustering data.* Prentice Hall.

Linde, Y., Buzo, A., & Gray, R. M. (1980). An algorithm for vector quantization design. *IEEE Transactions on Communications, 28*(1), 84-95.

Li, Tao & Ma, Sheng (2004). IFD: iterative feature and data clustering. In *Proceedings of SIAM International Conference on Data Mining*.

Li, Tao, Ma, Sheng & Ogihara, Mitsunori (2004a). Entropy-based criterion in categorical clustering. In *Proceedings of International Conference on Machine Learning (ICML 2004),* 536-543.

Li, Tao, Ma, Sheng & Ogihara, Mitsunori (2004b). Document clustering via adaptive subspace iteration. In *Proceedings of ACM SIGIR 2004,* 218-225.

Ng, A., Jordan, M., & Weiss, Y. (2001). On spectral clustering: analysis and an algorithm. In *Advances in Neural Information Processing Systems 14.*

Nishisato, S. (1980). *Analysis of categorical data: dual scaling and its applications.* Toronto: University of Toronto Press.

Slonim, N. and Tishby, N. (2000). Document clustering using word clusters via the information bottleneck method. In P*roceedings of ACM SIGIR 2000,* 208-215.

Tishby, N., Pereira, F.~C., & Bialek, W (1999). The information bottleneck method. In *Proceedings of the 37-th Annual Allerton Conference on Communication, Control and Computing,* 368-377.

Weiss, Y. (1999). Segmentation using eigenvectors: A unifying view. In *Proceedings of ICCV* (2). 975-982.

Zhang, T., Ramakrishnan, R. & Livny, M. (1996). BIRCH: an efficient data clustering method for very large databases. In *Proceedings of ACM SIGMOD Conference*, 103-114.

KEY TERMS

Cluster: A cluster is the set of entities that are *similar* between themselves and *dissimilar* to entities from other clusters.

Clustering: Clustering is the process of dividing the data into clusters.

Cluster Validation: Cluster validation evaluates the clustering results and judges the cluster structures

Co-clustering: Co-clustering performs simultaneous clustering of both points and their attributes by way of utilizing the canonical duality contained in the point-by-attribute data representation.

C**urse of Dimensionality:** The expression ``curse of dimensionality'' is due to Bellman and in statistics it relates to the fact that the convergence of any estimator to the true value of a smooth function defined on a space of high dimension is very slow. It has been used in various scenarios to refer to the fact that the complexity of learning grows significantly with the dimensions.

Spectral Clustering: Spectral clustering is the collection of techniques that perform clustering tasks using eigenvectors of matrices derived from the data.

Subspace Clustering: Subspace clustering is an extension of traditional clustering techniques that seeks to find clusters in different subspaces within a given dataset.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 176-179, copyright 2005 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

Comparing Four-Selected Data Mining Software

C

Richard S. Segall
Arkansas State University, USA

Qingyu Zhang
Arkansas State University, USA

INTRODUCTION

This chapter discusses four-selected software for data mining that are not available as free open-source software. The four-selected software for data mining are SAS® Enterprise Miner™, Megaputer PolyAnalyst® 5.0, NeuralWare Predict® and BioDiscovery GeneSight®, each of which was provided by partnerships with our university. These software are described and compared by their existing features, characteristics, and algorithms and also applied to a large database of forest cover types with 63,377 rows and 54 attributes. Background on related literature and software are also presented. Screen shots of each of the four-selected software are presented, as are future directions and conclusions.

BACKGROUND

Historical Background

Han and Kamber (2006), Kleinberg and Tardos (2005), and Fayyad et al. (1996) each provide extensive discussions of available algorithms for data mining.

Algorithms according to StatSoft (2006b) are operations or procedures that will produce a particular outcome with a completely defined set of steps or operations. This is opposed to heuristics that according to StatSoft (2006c) are general recommendations or guides based upon theoretical reasoning or statistical evidence such as "data mining can be a useful tool if used appropriately."

The Data Intelligence Group (1995) defined data mining as the extraction of hidden predictive information form large databases. According to The Data Intelligence Group (1995), "data mining tools scour databases for hidden patterns, finding predictive information that experts may miss because it lies outside their expectations."

Brooks (1997) describes rules-based tools as opposed to algorithms. Witten and Frank (2005) describe how data mining algorithms work including covering algorithms, instance-based learning, and how to use the WEKA, an open source data mining software that is a machine learning workbench.

Segall (2006) presented a chapter in the previous edition of this Encyclopedia that discussed microarray databases for biotechnology that included a extensive background on microarray databases such as that defined by Schena (2003), who described a microarray as "an ordered array of microscopic elements in a planar substrate that allows the specific binding of genes or gene products." The reader is referred to Segall (2006) for a more complete discussion on microarray databases including a figure on the overview of the microarray construction process.

Piatetsky-Shapiro (2003) discussed the challenges of data mining specific to microarrays, while Grossman et al. (1998) reported about three NSF (National Science Foundation) workshops on mining large massive and distributed data, and Kargupta at al. (2005) discussed the generalities of the opportunities and challenges of data mining.

Segall and Zhang (2004, 2005) presented funded proposals for the premises of proposed research on applications of modern heuristics and data mining techniques in knowledge discovery whose results are presented as in Segall and Zhang (2006a, 2006b) in addition to this chapter.

Software Background

There is a wealth of software today for data mining such as presented in American Association for Artificial Intelligence (AAAI) (2002) and Ducatelle (2006) for teaching data mining, Nisbet (2006) for CRM (Customer Relationship Management) and software review of Deshmukah (1997). StatSoft (2006a) presents screen shots of several softwares that are used for exploratory data analysis (EDA) and various data mining techniques. Proxeon Bioinformatics (2006) manufactures bioinformatics software for proteomics the study of protein and sequence information.

Lazarevic et al. (2006) discussed a software system for spatial data analysis and modeling. Leung (2004) compares microarray data mining software.

National Center for Biotechnology Information (NCBI) (2006) provides tools for data mining including those specifically for each of the following categories of nucleotide sequence analysis, protein sequence analysis and proteomics, genome analysis, and gene expression. Lawrence Livermore National Laboratory (LLNL) (2005) describes their Center for Applied Scientific Computing (CASC) that is developing computational tools and techniques to help automate the exploration and analysis of large scientific data sets.

MAIN THRUST

Algorithms of Four-Selected Software

This chapter specifically discusses four-selected data mining software that were chosen because these software vendors have generously offered their services and software to the authors at academic rates or less for use in both the classroom and in support of the two faculty summer research grants awarded as Segall and Zhang (2004, 2005).

SAS Enterprise Miner™ is a product of SAS Institute Inc. of Cary, NC and is based on the SEMMA approach that is the process of Sampling (S), Exploring (E), Modifying (M), Modeling (M), and Assessing (A) large amounts of data. SAS Enterprise Miner™ utilizes

Table 1. Description of data mining algorithms for PolyAnalyst® 5

Data Mining Algorithm	Underlying Algorithms
1. Discriminate	1. (a.) Fuzzy logic for classification 1. 1. (b.) Find Laws, PolyNet Predictor, or Linear Regression
2. Find Dependencies	2. ARNAVAC [See Key Terms]
3. Summary Statistics	3. Common statistical analysis functions
4. Link Analysis (LA)	4. Categorical, textual and Boolean attributes
5. Market and Transactional Basket Analysis	5. PolyAnalyst Market Basket Analysis
6. Classify	6. Same as that for Discriminate
7. Cluster	7. Localization of Anomalies Algorithm
8. Decision Forest (DF)	8. Ensemble of voting decision trees
9. Decision Tree	9. (a.) Information Gain splitting criteria (b.) Shannon information theory and statistical significance tests.
10. Find Laws	10. Symbolic Knowledge Acquisition Technology (SKAT) [See Key Terms]
11. Nearest Neighbor	11. PAY Algorithm
12. PolyNet Predictor	12. PolyNet Predictor Neural Network
13. Stepwise Linear Regression	13. Stepwise Linear Regression
14. Link Terms (LT)	14. Combination of Text Analysis and Link Analysis algorithms
15. Text Analysis (TA)	15.Combination of Text analysis algorithms augmented by statistical techniques
16. Text Categorization (TC)	16. Text Analysis algorithm and multiple subdivision splitting of databases.

a workspace with a drop-and-drag of icons approach to constructing data mining models.

SAS Enterprise Miner™ utilizes algorithms for decision trees, regression, neural networks, cluster analysis, and association and sequence analysis.

PolyAnalyst® 5 is a product of Megaputer Intelligence, Inc. of Bloomington, IN and contains sixteen (16) advanced knowledge discovery algorithms as described in Table 1 that was constructed using its User Manual by Megaputer Intelligence Inc. (2004; p. 163, p. 167, p.173, p.177, p. 186, p.196, p.201, p.207, p. 214, p. 221, p.226, p. 231, p. 235, p. 240, p.263, p. 274.).

NeuralWorks Predict® is a product of NeuralWare of Carnegie, PA. This software relies on neural networks, According to NeuralWare (2003, p.1):

"One of the many features that distinguishes Predict® from other empirical modeling and neural computing tools is that is automates much of the painstaking and time-consuming process of selecting and transforming the data needed to build a neural network."

NeuralWorks Predict® has a direct interface with Microsoft Excel that allows display and execution of the Predict® commands as a drop-down column within Microsoft Excel.

GeneSight™ is a product of BioDiscovery, Inc. of El Segundo, CA that focuses on cluster analysis using two main techniques of hierarchical and partitioning both of which are discussed in Prakash and Hoff (2002) for data mining of microarray gene expressions.

Both SAS Enterprise Miner™ and PolyAnalyst® 5 offer more algorithms than either GeneSight™ or NeuralWorks Predict®. These two software have algorithms for statistical analysis, neural networks, decision trees, regression analysis, cluster analysis, self-organized maps (SOM), association (e.g. market-basket) and sequence analysis, and link analysis. GeneSight™ offers mainly cluster analysis and NeuralWorks Predict® offers mainly neural network applications using statistical analysis and prediction to support these data mining results. PolyAnalyst® 5 is the only software of these that provides link analysis algorithms for both numerical and text data.

Applications of the Four-Selected Software to Large Database

Each of the four-selected software have been applied to a large database of forest cover type that is available on the same website of the Machine Learning Repository at the University of California at Irvine by Newman et al. (1998) for which results are shown in Segall and Zhang (2006a, 2006b) for different datasets of numerical abalone fish data and discrete nominal-valued mushroom data.

The forest cover type's database consists of 63,377 records each with 54 attributes that can be used to as inputs to predictive models to support decision-making processes of natural resource managers. The 54 columns of data are composed of 10 quantitative variables, 4 binary variables for wilderness areas, and 40 binary variables of soil types. The forest cover type's classes include Spruce-Fir, Lodgepole Pine, Ponderosa Pine, Cottonwood/Willow, Aspen, Douglas-Fir, Krummholz, and other.

The workspace of SAS Enterprise Miner™ is different than the other software because it uses icons that are user-friendly instead of only using spreadsheets of data. The workspace in SAS Enterprise Miner™ is constructed by using a drag-and-drop process from the icons on the toolbar which again the other software discussed do not utilize.

Figure 1 shows a screen shot of cluster analysis for the forest cover type data using

SAS Enterprise Miner™. From Figure 1 it can be seen using a slice with a standard deviation measurement, height of frequency, and color of the radius that this would yield only two distinct clusters: one with normalized mean of 0 and one with normalized mean of 1. If different measure of measurement, different slice height, and different key for color were selected than a different cluster figure would have resulted.

A screen shot of PolyAnalyst® 5.0 showing the input data of the forest cover type data with attribute columns of elevation, aspect, scope, horizontal distance to hydrology, vertical distance to hydrology, horizontal distance to roadways, hillshade 9AM, hillshade Noon, etc. PolyAnalyst® 5.0 yielded a classification probability of 80.19% for the forest cover type database.

Figure 2 shows the six significant classes of clusters corresponding to the six major cover types: Spruce-Fir, Lodgepole Pine, Ponderosa Pine, Cottonwood/Willow, Aspen, and Douglas-Fir. One of the results that can be seen from Figure 2 is that the most significant cluster for the aspect variable is the cover type of Douglas-Fir.

NeuralWare Predict® uses a Microsoft Excel spreadsheet interface for all of its input data and many of its outputs of computational results. Our research using NeuralWare Predict® for the forest type cover

Figure 1. SAS Enterprise Miner™ screen shot of cluster analysis

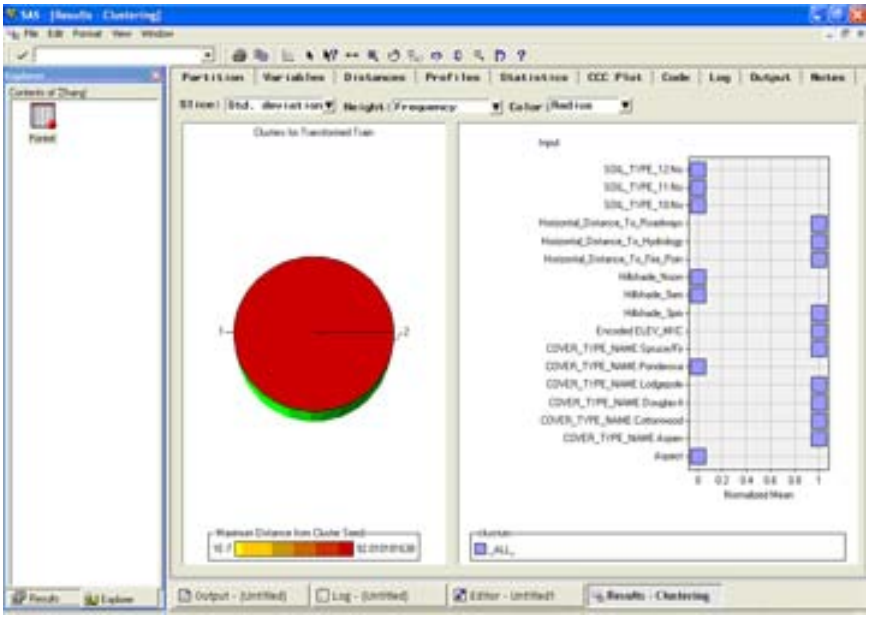
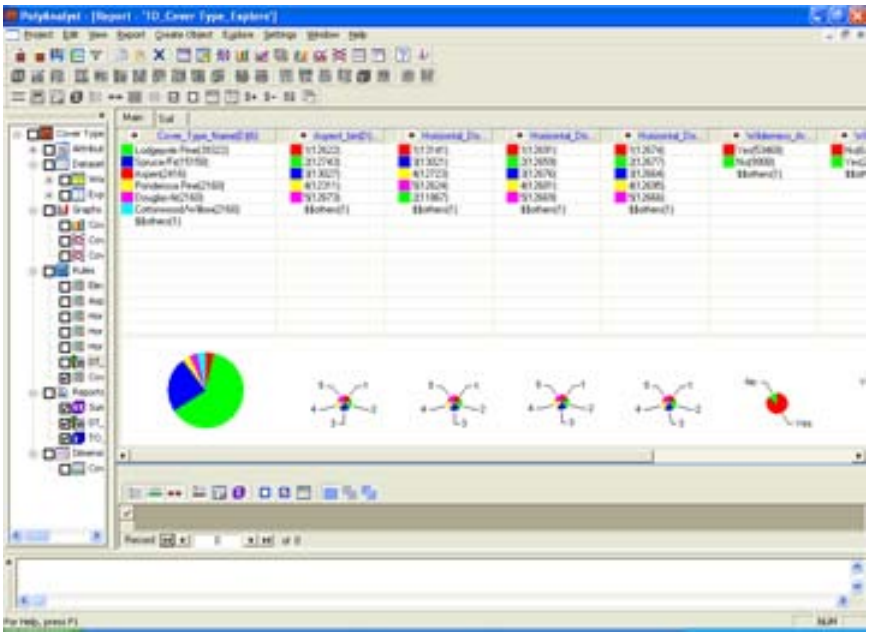


Figure 2. PolyAnalyst® 5.0 screen shot of cluster analysis



data indicates an accuracy of 70.6% with 70% of the sample for training and 30% of the sample for test in 2 minutes and 26 seconds of execution time.

Figure 3 is a screen shot of NeuralWare Predict® for the forest type cover data that indicates that an improved accuracy of 79.2% can be obtained using 100% of the 63,376 records for both training and testing in 3 minutes 21 seconds of execution time in evaluating model. This was done to investigate what could be the maximum accuracy that could be obtained

using NeuralWare Predict® for the same forest cover type database for comparison purposes to the other selected software.

Figure 4 is a screen shot using GeneSight® software by BioDiscovery Incorporated. That shows hierarchical clustering using the forest cover data set. As noted earlier, GeneSight® only performs statistical analysis and cluster analysis and hence no regression results for the forest cover data set can be compared with those of NeuralWare® Predict and PolyAnalyst®. It should be noted that from Figure 4 that the hierarchical clustering performed by GeneSight® for the forest cover type data set produced a multitude of clusters using the Euclidean distance metric.

FUTURE TRENDS

These four-selected software as described will be applied to a database that already has been collected of a different dimensionality. The database that has been presented in this chapter is of a forest cover type data set with 63,377 records and 54 attributes. The other database is a microarray database at the genetic level for a human lung type of cancer consisting of 12,600 records and 156 columns of gene types. Future simulations are to be performed for the human lung cancer data for each of the four-selected data mining software with their respective available algorithms and compared versus those obtained respectively for the larger database of 63,377 records and 54 attributes of the forest cover type.

CONCLUSION

The conclusions of this research include the fact that each of the software selected for this research has its own unique characteristics and properties that can be displayed when applied to the forest cover type database. As indicated, each software has it own set of algorithm types to which it can be applied. NeuralWare Predict® focuses on neural network algorithms, and Biodiscovery GeneSight® focuses on cluster analysis. Both SAS Enterprise Miner™ and Megaputer PolyAnalyst® employ each of the same algorithms except that SAS has a separate software SAS TextMiner® for text analysis. The regression results for the forest cover type data set are comparable for those obtained using NeuralWare Predict® and Megaputer PolyAnalyst®. The cluster analysis results for SAS Enterprise Miner™, Megaputer PolyAnalyst®, and Biodiscovery GeneSight® are unique to each software as to how they represent their results. SAS Enterprise Miner™ and NeuralWare Predict® both utilize Self-Organizing Maps (SOM) while the other two do not.

Figure 3. NeuralWare Predict® screen shot of complete neural network training (100%)

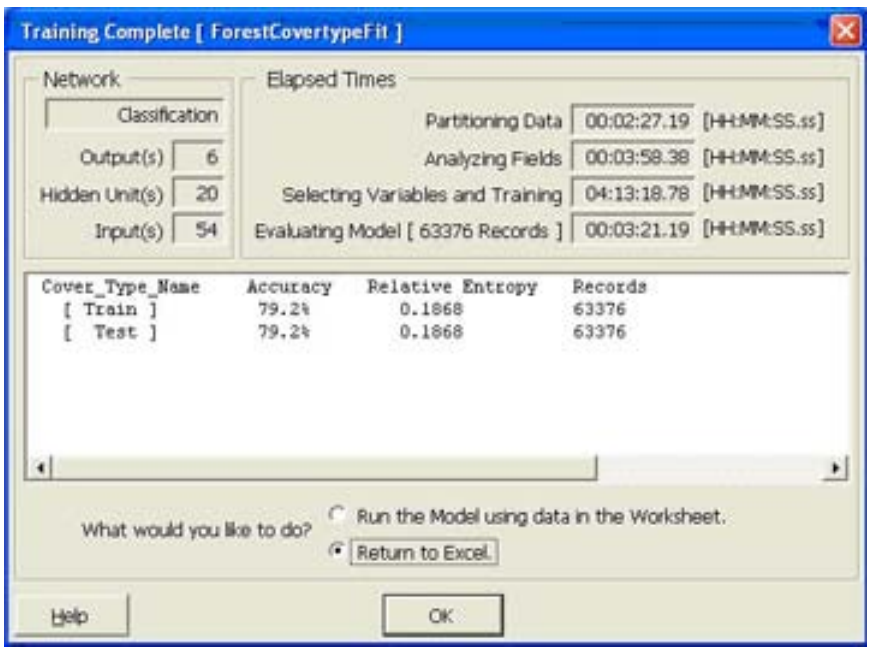
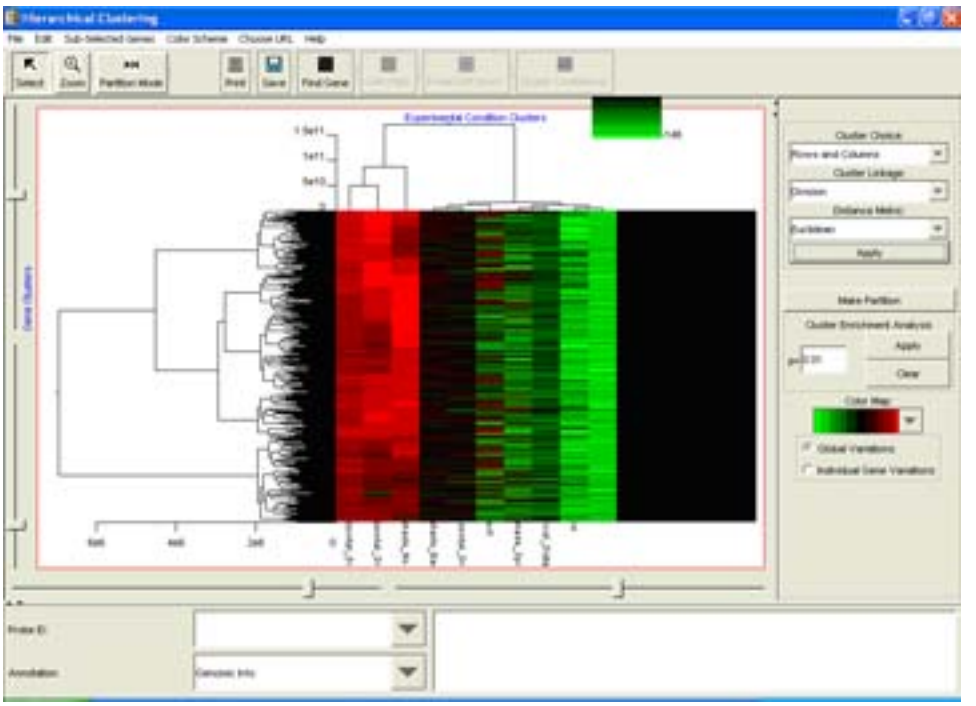


Figure 4. BioDiscovery GeneSight® screen shot of hierarchical clustering



The four-selected software can also be compared with respect to their cost of purchase. SAS Enterprise Miner™ is the most expensive and NeuralWare Predict® is the least expensive. Megaputer PolyAnalyst® and Biodiscovery GeneSight® are intermediate in cost to the other two software.

In conclusion, SAS Enterprise Miner™ and Megaputer PolyAnalyst® offer the greatest diversification of data mining algorithms.

ACKNOWLEDGMENT

The authors want to acknowledge the support provided by a 2006 Summer Faculty Research Grant as awarded to them by the College of Business of Arkansas State University. The authors also want to acknowledge each of the four software manufactures of SAS, Megaputer Intelligence, Inc., BioDiscovery, Inc., and NeuralWare, for their support of this research.

REFERENCES

AAAI (2002), American Association for Artificial Intelligence (AAAI) Spring Symposium on Information Refinement and Revision for Decision Making: Modeling for Diagnostics, Prognostics, and Prediction, Software and Data, retrieved from http://www.cs.rpi.edu/~goebel/ss02/software-and-data.html.

Brooks, P. (1997), Data mining today, *DBMS*, February 1997, retrieved from http://www.dbmsmag.com/9702d16.html.

Data Intelligence Group (1995), An overview of data mining at Dun & Bradstreet, *DIG White Paper 95/01*, retrieved from http://www.thearling.com.text/wp9501/wp9501.htm.

Deshmukah, A. V. (1997), Software review: ModelQuest Expert 1.0, *ORMS Today*, December 1997, retrieved from http://www.lionhrtpub.com/orms/orms-12-97/software-review.html.

Ducatelle, F., *Software for the data mining course, School of Informatics*, The University of Edinburgh, Scotland, UK, retrieved from http://www.inf.ed.ac.uk/teaching/courses/dme/html/software2.html.

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996), From Data Mining to Knowledge Discovery: An Overview. In *Advances in Knowledge Discovery and Data Mining*, eds. U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, 1–30. Menlo Park, Calif.: AAAI Press.

Grossman, R., Kasif, S., Moore, R., Rocke, D., & Ullman, J. (1998), *Data mining research: opportunities and challenges*, retrieved from http://www.rgrossman.com/epapers/dmr-v8-4-5.htm.

Han, J. & Kamber, M. (2006), *Data Mining: Concepts and Techniques*, 2nd edition, Morgan Kaufman, San Francisco, CA.

Kargupta, H., Joshi, A., Sivakumar, K., & Yesha, Y. (2005), *Data mining: Next generation challenges and future directions*, MIT/AAAI Press, retrieved from http:///www.cs.umbc.edu/~hillol/Kargupta/ngdmbook.html.

Kleinberg, J. & Tardos, E., (2005), *Algorithm Design*, Addison-Wesley, Boston, MA.

Lawrence Livermore National Laboratory (LLNL), The Center for Applied Scientific Computing (CASC), *Scientific data mining and pattern recognition: Overview*, retrieved from http://www.llnl.gov/CASC/sapphire/overview/html.

Lazarevic A., Fiea T., & Obradovic, Z., *A software system for spatial data analysis and modeling*, retrieved from http://www.ist.temple.edu?~zoran/papers/lazarevic00.pdf.

Leung, Y. F.(2004), *My microarray software comparison – Data mining software*, September 2004, Chinese University of Hong Kong, retrieved from http://www.ihome.cuhk.edu.hk/~b400559/arraysoft mining specific.html.

Megaputer Intelligence Inc. (2004), *PolyAnalyst 5 Users Manual*, December 2004, Bloomington, IN 47404.

Megaputer Intelligence Inc. (2006), *Machine learning algorithms*, retrieved from http://www.megaputer.com/products/pa/algorithms/index/php3.

Moore, A., *Statistical data mining tutorials*, retrieved from http://www.autonlab.org/tutorials.

National Center for Biotechnology Information (2006), National Library of Medicine, National Institutes of Heath, *NCBI tools for data mining*, retrieved from http://www.ncbi.nlm,nih.gov/Tools/.

NeuralWare (2003), NeuralWare Predict® The complete solution for neural data modeling: *Getting Started Guide for Windows*, NeuralWare, Inc., Carnegie, PA 15106

Newman, D.J. & Hettich, S. & Blake, C.L. & Merz, C.J. (1998). *UCI Repository of machine learning databases*, Irvine, CA: University of California, Department of Information and Computer Science, http://www.ics.uci.edu/~mlearn/MLRepository.html

Nisbet, R. A. (2006), Data mining tools: Which one is best for CRM? Part 3, *DM Review*, March 21, 2006, retrieved from http://www.dmreview.com/editorial/dmreview/print_action.cfm?articleId=1049954.

Piatetsky-Shapiro, G. & Tamayo, P. (2003), Microarray data mining: Facing the challenges, *SIGKDD Exploration*, vo.5, n.2, pages 1-5, December, retrieved from http://portal.acm.org/citation.cfm?doid=980972.980974 and http://www.broad.mit.edu/mpr/publications/projects/genomics/Microarray_data_mining_facing%20_the_challenges.pdf.

Prakash, P. & Hoff, B. (2002) *Microarray gene expression data mining with cluster analysis using GeneSight™*, Application Note GS10, BioDiscovery, Inc., El Segundo, CA, retrieved from http://www.biodiscovery.com/index/cms-filesystem-action?file=AppNotes-GS/appnotegs10.pdf.

Proxeon Bioinformatics, *Bioinformatics software for proteomics, from proteomics data to biological sense in minutes*, retrieved from http://www.proxeon.com/protein-sequence-databases-software.html.

SAS® Enterprise Miner™, SAS Incorporated, Cary, NC, retrieved from http://www.sas.com/technologies/analytics/datamining/miner.

Schena, M. (2003*). Microarray analysis*, New York, John Wiley & Sons, Inc.

Segall, R.S. (2006), Microarray databases for biotechnology, *Encyclopedia of Data Warehousing and Mining*, John Wang, Editor, Idea Group, Inc., pp. 734-739.

Segall, R. S, & Zhang, Q. (2004). *Applications of modern heuristics and data mining techniques in knowledge discovery*, funded proposal submitted to Arkansas State University College of Business Summer Research Grant Committee.

Segall, R. S. & Zhang, Q. (2005). *Continuation of research on applications of modern heuristics and data mining techniques in knowledge discovery*, funded proposal submitted to Arkansas State University College of Business Summer Research Grant Committee.

Segall, R.S. & Zhang, Q. (2006a). Applications of neural network and genetic algorithm data mining techniques in bioinformatics knowledge discovery – A preliminary study, *Proceedings of the Thirty-seventh Annual Conference of the Southwest Decision Sciences Institute*, Oklahoma City, OK, v. 37, n. 1, March 2-4, 2006.

Segall, R. S. & Zhang, Q. (2006b). Data visualization and data mining of continuous numerical and discrete nominal-valued microarray databases for biotechnology, *Kybernetes: International Journal of Systems and Cybernetics*, v. 35, n. 9/10.

StatSoft, Inc. (2006a). *Data mining techniques*, retrieved from http://www.statsoft.com/textbook/stdatmin.html.

StatSoft, Inc. (2006b). *Electronic textbook*, retrieved from http://www.statsoft.com/textbook/glosa.html.

StatSoft, Inc. (2006c). *Electronic textbook*, retrieved from http://www.statsoft.com/textbook/glosh.html.

Tamayo, P. & Ramaswamy, S. (2002). *Cancer genomics and molecular pattern recognition*, Cancer Genomics Group, Whitehead Institute, Massachusetts Institute of Technology, retrieved from http://www.broad.mit.edu/mpr/publications/projects/genomics/Humana_final_Ch_06_23_2002%20SR.pdf.

Witten, IH & Frank E. (2005). *Data mining: Practical machine learning tools and techniques with Java implementation*, Morgan Kaufman.

KEY TERMS

Algorithm: That which produces a particular outcome with a completely defined set of steps or operations.

ARNAVAC: An underlying machine language algorithm used in PolyAnalyst® for the comparison of the target variable distributions in approximately homogeneously equivalent populated multidimensional hyper-cubes.

Association and sequence analysis: A data mining method that relates first a transaction and an item and secondly also examines the order in which the products are purchased.

BioDiscovery GeneSight®: A program for efficient data mining, visualization, and reporting tool that can analyze massive gene expression data generated by microarray technology.

Data Mining: The extraction of interesting and potentially useful information or patterns from data in large databases; also known as Knowledge Discovery in Data (KDD).

Link Analysis (LA): A technique used in data mining that reveals and visually represents complex patterns between individual values of all categorical and Boolean attributes.

Link Terms (LT): A technique used in text mining that reveals and visually represents complex patterns of relations between terms in textual notes.

Market and transactional basket analysis: Algorithms that examine a long list of transactions to determine which items are most frequently purchased together, as well as analysis of other situations such as identifying those sets of questions of a questionnaire that are frequently answered with the same categorical answer.

Megaputer PolyAnalyst® 5.0: A powerful multistrategy data mining system that implements a broad variety of mutually complementing methods for the automatic data analysis.

Microarray Databases: Store large amounts of complex data as generated by microarray experiments (e.g. DNA)

NeuralWare NeuralWorks Predict®: A software package that integrates all the capabilities needed to apply neural computing to a wide variety of problems.

SAS® Enterprise Miner™: Software that that uses an drop-and-drag object oriented approach within a workspace to performing data mining using a wide variety of algorithms.

Symbolic Knowledge Acquisition Technology (SKAT): An algorithm developed by Megaputer Intelligence and used in PolyAnalyst® 5.0 that uses methods of evolutionary programming for high-degree rational expressions that can efficiently represent nonlinear dependencies.

WEKA: Open-source data mining software that is a machine learning workbench.

Compression-Based Data Mining

Eamonn Keogh
University of California - Riverside, USA

Li Wei
Google, Inc., USA

John C. Handley
Xerox Innovation Group, USA

INTRODUCTION

Compression-based data mining is a universal approach to clustering, classification, dimensionality reduction, and anomaly detection. It is motivated by results in bioinformatics, learning, and computational theory that are not well known outside those communities. It is based on an easily computed compression dissimilarity measure (CDM) between objects obtained by compression. The basic concept is easy to understand, but its foundations are rigorously formalized in information theory. The similarity between any two objects (XML files, time series, text strings, molecules, etc.) can be obtained using a universal lossless compressor. The compression dissimilarity measure is the size of the compressed concatenation of the two objects divided by the sum of the compressed sizes of each of the objects. The intuition is that if two objects are similar, lossless compressor will remove the redundancy between them and the resulting size of the concatenated object should be close the size of the larger of the two compressed constituent objects. The larger the CDM between two objects, the more dissimilar they are.

Classification, clustering and anomaly detection algorithms can then use this dissimilarity measure in a wide variety of applications. Many of these are described in (Keogh et al., 2004), (Keogh et al. 2007), and references therein. This approach works well when (1) objects are large and it is computationally expensive to compute other distances (e.g., very long strings); or (2) there are no natural distances between the objects or none that are reasonable from first principles. CDM is "parameter-free" and thus avoids over-fitting the data or relying upon assumptions that may be incorrect (Keogh et al., 2004).

CDM enjoys the following properties:

1. Because it makes no distributional or modeling assumptions about the data, it allows true exploratory data mining.
2. The accuracy of CDM is often greatly superior to those of parameter-laden or model-based algorithms, even if we allow these algorithms to search exhaustively over their parameter spaces.
3. CDM uses compression algorithms which are typically space and time efficient. As a consequence, CDM can be much more efficient than other algorithms, in some cases by three or four orders of magnitude.
4. CDM makes no assumption about the format of the data, nor does it require extensive data cleaning to be effective.

BACKGROUND

The use of data compression to classify sequences is also closely related to the Minimum Description Length (MDL) and Minimum Message Length (MML) principles (Grünwald, 2007), (Wallace, 2005). See keyword definitions at the end of the article. The MDL/MML principle has generated an extensive body of literature in the data mining community. CDM is a related concept, but it requires no probabilistic concepts and can be universally applied.

CDM is based on the concept of Kolmogorov complexity, a measure of randomness of strings based on their information content (Li & Vitanyi, 1997). It was proposed by Kolmogorov in 1965 to quantify the randomness of strings and other objects in an objective

and absolute manner. The Kolmogorov complexity K(x) of a string x is defined as the length of the shortest program capable of producing x on a universal computer — such as a Turing machine. Different programming languages will give rise to distinct values of K(x), but one can prove that the differences are only up to a fixed additive constant. Intuitively, K(x) is the minimal quantity of information required to generate x by an algorithm. The conditional Kolmogorov complexity K(x|y) of x to y is defined as the length of the shortest program that computes x when y is given as an auxiliary input to the program. The function K(xy) is the length of the shortest program that outputs y concatenated to x. In Li et al. (2001), the authors consider the distance between two strings, x and y, defined as

$$d_k(x,y) = \frac{K(x \mid y) + K(y \mid x)}{K(xy)} \quad (1)$$

which satisfies the triangle inequality, up to a small error term. A more mathematically precise distance was proposed in Li et al. (2003). Kolmogorov complexity is without a doubt the ultimate lower bound among all measures of information content. Unfortunately, it cannot be computed in the general case (Li and Vitanyi, 1997). As a consequence, one must approximate this distance. It is easy to realize that universal compression algorithms give an upper bound to the Kolmogorov complexity. In fact, K(x) is the best compression that one could possibly achieve for the text string x. Given a data compression algorithm, we define C(x) as the size of the compressed size of x, C(xy) as the size of the compressed size of the concatenation of x and y and C(x|y) as the compression achieved by first training the compression on y, and then compressing x. For example, if the compressor is based on a textual substitution method, one could build the dictionary on y, and then use that dictionary to compress x. We can approximate (1) by the following distance measure

$$d_c(x,y) = \frac{C(x \mid y) + C(y \mid x)}{C(xy)} \quad (2)$$

The better the compression algorithm, the better the approximation of d_c for d_k is. Li et al., (2003) have shown that d_c is a similarity metric and can be successfully applied to clustering DNA and text. However, the measure would require hacking the chosen compression algorithm in order to obtain C(x|y) and C(y|x). CDM simplifies this distance even further. In the next section, we will show that a simpler measure can be just as effective.

A comparative analysis of several compression-based distances has been recently carried out in Sculley and Brodley (2006). The idea of using data compression to classify sequences over finite alphabets is not new. For example, in the early days of computational biology, lossless compression was routinely used to classify and analyze DNA sequences. Refer to, e.g., Allison et al. (2000), Baronchelli et al. (2005), Farach et al. (1995), Frank et al. (2000), Gatlin (1972), Kennel (2004), Loewenstern and Yianilos (1999), Needham and Dowe (2001), Segen (1990), Teahan et al. (2000), Ferragina et al. (2007), Melville et al. (2007) and references therein for a sampler of the rich literature existing on this subject. More recently, Benedetto et al. (2002) have shown how to use a compression-based measure to classify fifty languages. The paper was featured in several scientific (and less-scientific) journals, including Nature, Science, and Wired.

MAIN FOCUS

CDM is quite easy to implement in just about any scripting language such as Matlab, Perl, or R. All that is required is the ability to programmatically execute a lossless compressor, such as gzip, bzip2, compress, WinZip and the like and store the results in an array. Table 1 shows the complete Matlab code for the compression-based dissimilarity measure.

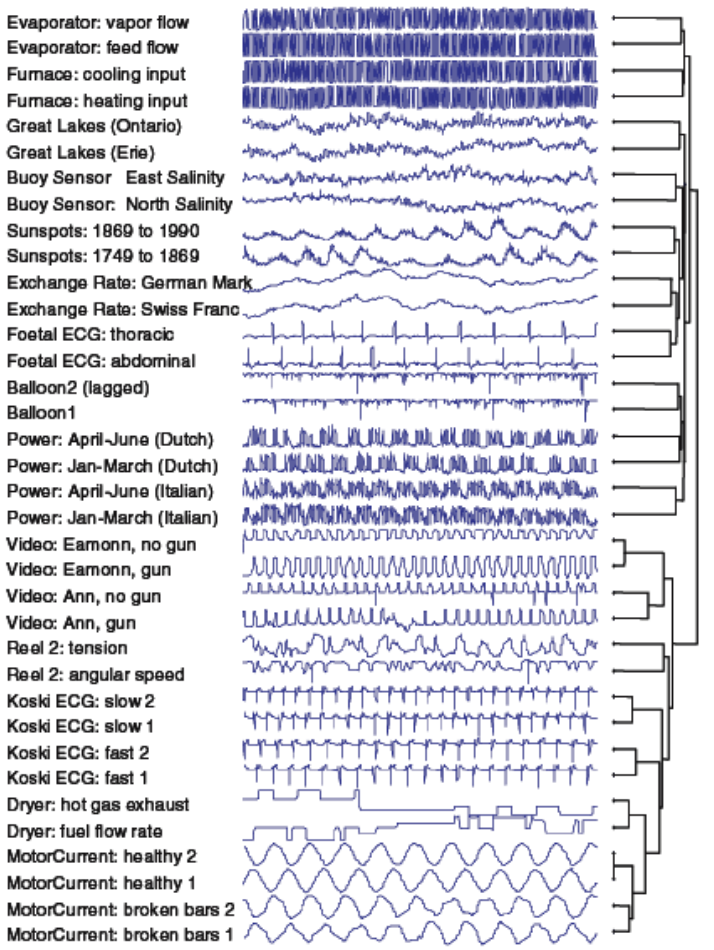
Once pairwise dissimilarities have been computed between objects, the dissimilarity matrix can be used for clustering (e.g. hierarchical agglomerative clustering), classification (e.g., k-nearest neighbors), dimensionality reduction (e.g., multidimensional scaling), or anomaly detection.

The best compressor to capture the similarities and differences between objects is the compressor that compresses the data most. In practice, one of a few easily obtained lossless compressors works well and the best one can be determined by experimentation. In some specialized cases, a lossless compressor designed for the data type provides better results (e.g., DNA clustering, Benedetto et al. 2002). The theoretical relationship between optimal compression and features for clustering is the subject of future research.

Table 1. Matlab code for compression dissimilarity measure

```
function Dist = CDM(A,B)
save A.txtA–ASCII                                          % Save variable A as A.txt
zip('A.zip', 'A.txt');                                     % Compress A.txt
A_file = dir('A.zip');                                     % Get file information
save B.txt B –ASCII                                        % Save variable B as B.txt
zip('B.zip', 'B.txt');                                     % Compress B.txt
B_file = dir('B.zip');                                     % Get file information
A_n_B = [A; B];                                            % Concatenate A and B
save A_n_B.txt A_n_B –ASCII                                % Save A_n_B.txt
zip('A_n_B.zip', 'A_n_B.txt');                             % Compress A_n_B.txt
A_n_B_file = dir('A_n_B.zip');                             % Get file information
dist = A_n_B_file.bytes / (A_file.bytes + B_file.bytes);   % Return CDM dissimilarity
```

Figure 1. Thirty-six time series (in 18 pairs) clustered using CDM.



To illustrate the efficacy of CDM, consider three examples: two experiments and a real-world application which was unanticipated by researchers in this field.

Time Series

The first experiment examined the UCR Time Series Archive (Keogh and Folias, 2002) for datasets that come in pairs. For example, in the Foetal-ECG dataset, there are two time series, thoracic and abdominal, and in the Dryer dataset, there are two time series, hot gas exhaust and fuel flow rate. There are 18 such pairs, from a diverse collection of time series covering the domains of finance, science, medicine, industry, etc.

While the correct hierarchical clustering at the top of a classification tree is somewhat subjective, at the lower level of the tree, we would hope to find a single bifurcation separating each pair in the dataset. Our metric, Q, for the quality of clustering is therefore the number of such correct bifurcations divided by 18, the number of datasets. For a perfect clustering, Q = 1, and because the number of dendrograms of 36 objects is greater than 3×10^{49}, for a random clustering, we would expect $Q \approx 0$.

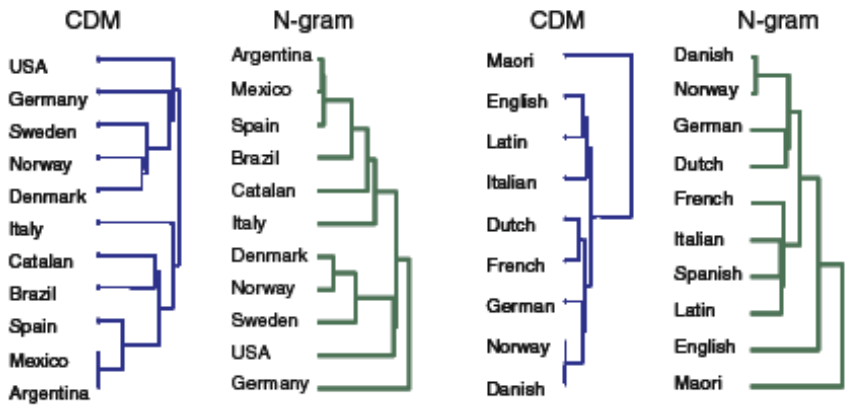
For many time series distance/dissimilarity/similarity measures that has appeared in major journals within the last decade, including dynamic time warping and longest common subsequence, we used hierarchical agglomerative clustering with single linkage, complete linkage, group average linkage, and Wards methods, and reported only the best performing results (Keogh et al., 2007). Figure 1 shows the resulting dendrogram for CDM, which achieved a perfect clustering with Q = 1. Although the higher level clustering is subjective, CDM seems to do very well. For example, the appearance of the Evaporator and Furnace datasets in the same subtree is quite intuitive, and similar remarks can be made for the two Video datasets and the two Motor-Current datasets. For more details on this extensive experiment, see (Keogh et al., 2007).

Natural Language

A similar experiment is reported in Benedetto et al. (2002). We clustered the text of various countries' Yahoo portals, only considering the first 1,615 characters, which is the size of the smallest webpage (excluding white spaces). Figure 2 (left) shows the clustering result. Note that the first bifurcation correctly divides the tree into Germanic and Romance languages.

The clustering shown is much better than that achieved by the ubiquitous cosine similarity measure

Figure 2 (Left) The clustering achieved by our approach and n-gram cosine measure (Rosenfeld, 2000) on the text from various Yahoo portals (January 15, 2004). The smallest webpage had 1,615 characters, excluding white spaces. (Right) The clustering achieved by our approach on the text from the first 50 chapters of Genesis. The smallest file had 132,307 characters, excluding white spaces. Maori, a Malayo–Polynesian language, is clearly identified as an "outlier."



used at the word level. In retrospect, this is hardly surprising. Consider the following English, Norwegian, and Danish words taken from the Yahoo portals:

English: {England, information, addresses}
Norwegian: {Storbritannia, informasjon, adressebok}
Danish: {Storbritannien, informationer, adressekartotek}

Because there is no single word in common to all (even after stemming), the three vectors are completely orthogonal to each other. Inspection of the text is likely to correctly conclude that Norwegian and Danish are much more similar to each other than they are to English. CDM can leverage off the same cues by finding repeated structure within and across texts. To be fairer to the cosine measure, we also tried an n-gram approach at the character level. We tested all values of N from 1 to 5, using all common linkages. The results are reported in Figure 2. While the best of these results are produces an intuitive clustering, other settings do not. We tried a similar experiment with text from various translations of the first 50 chapters of the Bible, this time including what one would expect to be an outlier, the Maori language of the indigenous people of New Zealand. As shown in Figure 2 (right) the clustering is correct, except for an inclusion of French in the Germanic subtree.

This result does not suggest that CDM replaces the vector space model for indexing text or a linguistic aware method for tracing the evolution of languages – it simply shows that for a given dataset which we know nothing about, we can expect CDM to produce reasonable results that can be a starting point for future study.

Questionnaires

CDM has proved successful in clustering semi-structured data not amenable to other methods and in an application that was unanticipated by its proponents. One of the strengths of Xerox Corporation sales is it comprehensive line of products and services. The sheer diversity of its offerings makes it challenging to present customers with compelling configurations of hardware and software. The complexity of this task is reduced by using a configuration tool to elicit responses from a customer about his or her business and requirements. The questionnaire is an online tool with which sales associates navigate based on customer responses. Different questions are asked based on how previous questions are answered. For example, if a customer prints only transactions (e.g., bills), a different sequence of questions is presented than if the customer prints annual reports with color content. The resulting data is a tree-structured responses represented in XML as shown in Figure 3.

As set of rules was hand-crafted to generate a configuration of products and feature options given the responses. What is desired was a way to learn which configurations are associated with which responses. The first step is to cluster the responses, which we call case logs, into meaningful groups. This particular "semi-structured" data type is not amenable to traditional text clustering, neither by a vector space approach like latent semantic indexing (Deerwester et al 1990) nor by

Figure 3. A sample case log fragment.

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<GUIJspbean>
<GUIQuestionnaireQuestionnaireVector>
…
<GUIQuestionnaireLocalizableMessage>Printing Application Types (select all that apply)
</GUIQuestionnaireLocalizableMessage>
<GUIQuestionnaireSelectMultipleChoice isSelected="false">
<GUIQuestionnaireLocalizableMessage>General Commercial / Annual Reports
</GUIQuestionnaireLocalizableMessage>
</GUIQuestionnaireSelectMultipleChoice>
<GUIQuestionnaireSelectMultipleChoice isSelected=»false»>
<GUIQuestionnaireLocalizableMessage>Books</GUIQuestionnaireLocalizableMessage>
</GUIQuestionnaireSelectMultipleChoice>
…
</GUIQuestionnaireQuestionnaireVector>
<GUIJspbeanAllWorkflows>
<OntologyWorkflowName>POD::Color Split::iGen3/Nuvera::iWay</OntologyWorkflowName>
…
</GUIJspbeanAllWorkflows>
</GUIJspbean>
```

a probabilistic generative model such as probabilistic latent semantic analysis (Hoffmann, 2001). This because frequencies of terms (responses such as 'yes' or 'no') carry little information – the information is contained in which questions were asked and in what order. String (or graph) edit distance would seem to be another appropriate choice. However, the logs, even with many XML tags stripped, are 10-24 kilobytes in size, posing computational problems. Due to the inapplicability of the classic distance measurements, we used CDM for our clustering task. Based on 200 logs, a CDM-based approach using hierarchical agglomerative clustering resulted in clusters that were meaningful (a pleasant surprise for a purely algorithmic approach with no encoded domain knowledge). The clusters were used to build a nearest neighbor classification system with errors of 4%. See Wei et al. (2006) for details.

FUTURE TRENDS

Although a rich and diverse set of applications is emerging, a rigorous theoretical analysis of compression-based dissimilarity measure has yet to be done. The interplay between feature extraction and compression is not fully understood.

The compression-based dissimilarity measure could be modified to be a distance measure, or better still, a distance metric in which the distance between two identical objects is zero and the triangle inequality holds, we could avail of a wealth of pruning and indexing techniques to speed up classification Ratanamahatana and Keogh (2004), clustering Elkan (2003), and similarity search Vlachos et al. (2003). While it is unlikely that CDM can be transformed in a true metric, it may be possible to prove a weaker version of the triangular inequality, which can be bounded and used to prune the search space (Elkan 2003). A step in this direction is demonstrated in the recent paper (Sculley and Brodley 2006).

Finally, we note that our approach is clearly not suitable for classifying or clustering low dimensionality data (although Figure 2 shows exceptionally good results on time series with only 1,000 data points).We plan to theoretically and empirically investigate the limitations on object sizes that we can meaningfully work with using our proposed approach.

CONCLUSION

CDM produces interpretable and accurate clustering and classification results across a wide variety of data types and serves as an easy-to-use first try at data exploration. It often works when no natural metrics are available or the computational complexity of an appropriate method is too great (e.g., an edit distance between very long strings). We expect the body of successful applications to continue to grow.

REFERENCES

Allison, L., Stern, L., Edgoose, T., & Dix, T. I. (2000). Sequence complexity for biological sequence analysis. *Computational Chemistry*, 24(1), 43–55.

Baronchelli, A., Caglioti, E., & Loreto, V. (2005). Artificial sequences and complexity measures. *Journal of Statistical Mechanics: Theory and Experiment*, 4, P04002.

Benedetto, D., Caglioti, E., Loreto, V. (2002). Language trees and zipping. *Physical Review Letters*, 88, 048702

Deerwester, S., S. T. Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing *by latent semantic analysis. Journal* of the American Society for Information Science, 41, 391-407.

Elkan, C. (2003). Using the triangle inequality to accelerate k-means. In *Proceedings of ICML 2003* (pp. 147–153). Washington DC.

Farach, M., Noordewier, M., Savari, S., Shepp, L., Wyner, A., & Ziv, J. (1995). On the entropy of DNA: algorithms and measurements based on memory and rapid convergence. In *Proceedings of the Symposium on Discrete Algorithms* (pp. 48-57). San Francisco: CA.

Ferragina, P., Giancarlo, R., Greco, V. , Manzini, G., & Valiente, G. (2007). Compression-based classification of biological sequences and structures via the Universal Similarity Metric: experimental assessment. *Bioinformatics*, 8(1), 252 – 271.

Frank, E., Chui, C., & Witten, I. (2000). Text categorization using compression models. In *Proceedings of the IEEE Data Compression Conference* (p. 555). Snowbird: Utah: IEEE Press.

Gatlin, L. (1972). *Information theory and the living systems*. Columbia University Press.

Grünwald, P. D. (2007). *The Minimum Description Length Principle*. MIT Press.

Hofmann, T., (2001). Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning Journal*, 42(1), 177 – 196.

Kennel, M. (2004). Testing time symmetry in time series using data compression dictionaries. *Physical Review E*, 69, 056208.

Keogh, E. & Folias, T. (2002). The UCR time series data mining archive. University of California, Riverside CA [http://www.cs.ucr.edu/eamonn/TSDMA/index.html]

Keogh, E., Lonardi, S. and Ratanamahatana, C., (2004) Towards parameter-free data mining. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 206-215). Seattle, WA.

Keogh, E., Lonardi, S. and Ratanamahatana, C., Wei, L., Lee, S-H., & Handley, J. (2007) Compression-based data mining of sequential data. *Data Mining and Knowledge Discovery*, 14(1), 99-129.

Li, M., Badger, J. H., Chen. X., Kwong, S., Kearney. P., & Zhang, H. (2001). An information-based sequence distance and its application to whole mitochondrial genome phylogeny. *Bioinformatics*, 17,149–154.

Li, M., Chen,X., Li, X., Ma, B., & Vitanyi, P. (2003). The similarity metric. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms* (pp 863–872). Baltimore: MD.

Li, M. & Vitanyi, P. (1997). *An introduction to Kolmogorov Complexity and its Applications*, 2nd ed., Springer Verlag: Berlin.

Loewenstern, D. & Yianilos, P. N. (1999). Significantly lower entropy estimates for natural DNA sequences. *Journal of Computational Biology*, 6(1), 125-142.

Melville, J., Riley, J., & Hirst, J. (2007) Similarity by compression. *Journal of Chemical Information and Modeling*, 47(1), 25-33.

Needham, S. & Dowe, D. (2001). Message length as an effective Ockham's razor in decision tree induction, In *Proceedings of the Eighth International Workshop on AI and Statistics* (pp. 253–260), Key West: FL.

Ratanamahatana, C. A. & Keogh, E. (2004). Making time-series classification more accurate using learned constraints. In *Proceedings of SIAM international Conference on Data Mining (SDM '04)*, (pp. 11-22), Lake Buena Vista: Florida

Rosenfeld, R. (2000). Two decades of statistical language modeling: where do we go from here? *Proceedings of the IEEE*, 88(8), 1270 – 1278.

Sculley, D., & Brodley, C. E. (2006). Compression and machine learning: a new perspective on feature space vectors. In *Proceedings of IEEE Data Compression Conference* (pp 332–341), Snowbird:UT: IEEE Press.

Segen, J. (1990). Graph clustering and model learning by data compression. In *Proceedings of the Machine Learning Conference* (pp 93–101), Austin:TX

Teahan,. W. J, Wen, Y., McNab, R. J., & Witten, I. H. (2000) A compression-based algorithm for Chinese word segmentation. *Computational Linguistics*, 26, 375–393.

Vlachos, M., Hadjieleftheriou, M., Gunopulos, D., & Keogh, E. (2003). Indexing multi-dimensional time series with support for multiple distance measures. In *Proceedings of the 9th ACM SIGKDD* (pp 216–225). Washington, DC.

Wallace, C. S. (2005). Statistical *and Inductive Inference by Minimum Message Length*. Springer.

Wei, L., Handley, J., Martin, N., Sun, T., & Keogh, E., (2006). Clustering workflow requirements using compression dissimilarity measure., In *Sixth IEEE International Conference on Data Mining - Workshops* (ICDMW'06) (pp. 50-54). IEEE Press.

KEY TERMS

Anomaly Detection: From a set of normal behaviors, detect when an observed behavior is unusual or abnormal, often by using distances between behaviors and detecting unusually large deviations.

Compression Dissimilarity Measure: Given two objects x and y and a universal, lossless compressor

C, the compression dissimilarity measure is the compressed size C(xy) of the concatenated objects divided by the sum C(x) + C(y) of the sizes of the compressed individual objects.

Kolmogorov Complexity: The Kolmogorov complexity of a string x of symbols from a finite alphabet is defined as the length of the shortest program capable of producing x on a universal computer such as a Turing machine.

Lossless Compression: A means of data compression such that the data can be encoded and decoded in its entirety with no loss of information or data. Examples include Lempel-Ziv encoding and its variants.

Minimum Description Length: The Minimum Description Length principle for modeling data states that the best model, given a limited set of observed data, is the one that can be describes the data as succinctly as possible, or equivalently that delivers the greatest compression of the data.

Minimum Message Length: The Minimum Message Length principle, which is formalized in probability theory, holds that given models for a data set, the one generating the shortest message that includes both a description of the fitted model and the data is probably correct.

Multidimensional Scaling: A method of visualizing high dimensional data in fewer dimensions. From a matrix of pairwise distances of (high dimensional) objects, objects are represented in a low dimensional Euclidean space, typically a plane, such that the Euclidean distances approximate the original pairwise distances as closely as possible.

Parameter-Free Data Mining: An approach to data mining that uses universal lossless compressors to measure differences between objects instead of probability models or feature vectors.

Universal Compressor: A compression method or algorithm for sequential data that is lossless and uses no model of the data. Performs well on a wide class of data, but is suboptimal for data with well-defined statistical characteristics.

Computation of OLAP Data Cubes

Amin A. Abdulghani
Data Mining Engineer, USA

INTRODUCTION

The focus of online analytical processing (OLAP) is to provide a platform for analyzing data (e.g., sales data) with multiple dimensions (e.g., product, location, time) and multiple measures (e.g., total sales or total cost). OLAP operations then allow viewing of this data from a number of perspectives. For analysis, the object or data structure of primary interest in OLAP is a data cube. A detailed introduction to OLAP is presented in (Han & Kambler, 2006).

BACKGROUND

Consider a 3-D cube model shown in Figure 1 representing sales data. It has three dimensions year, product and location. The measurement of interest is total sales. In olap terminology, since this cube models the base data, it forms a 3-D *base cuboid*. A *cuboid* in general is a group-by of a subset of dimensions of the base data, obtained by aggregating all tuples on these dimensions. So, for example for our sales data we have three 2-d cuboids namely (year, product), (product, location) and (year, location), three 1-d cuboids (year), (location) and (product) and one 0-d cuboid in which aggregation is performed on the whole data. A cuboid which is not a base cuboid is called an *aggregate cuboid* while a 0-D cuboid is called an *apex cuboid*. For base data, with n diemensions, the union of of all *k*-dimensional (k<=n) cuboids forms *an n-dimensional data cube*. A *cell* represents an association of a measure *m* (e.g., total sales) with a member of every dimension in a cuboid e.g. C1(product="toys", location="NJ", year="2004"). The dimensions not present in the cell are aggregated over all possible members. For example, you can have a two-dimensional (2-D) cell, C2(product="toys", year="2004"). Here, the implicit value for the dimension location is '*', and the measure *m* (e.g., total sales) is aggregated over all locations. Any of the standard

Figure 1. A 3-D base cuboid with an example 3-D cell.

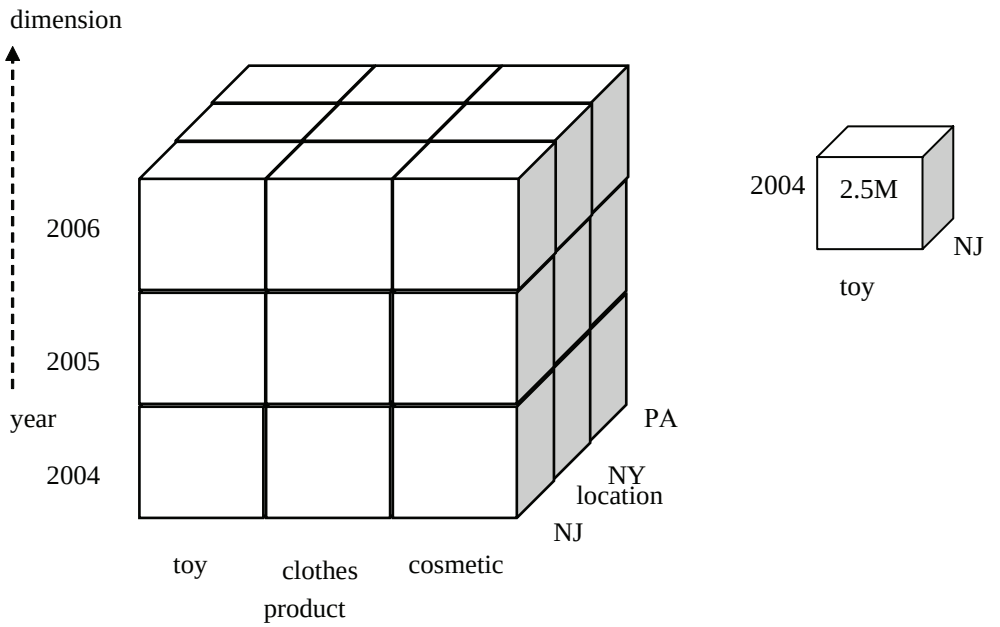
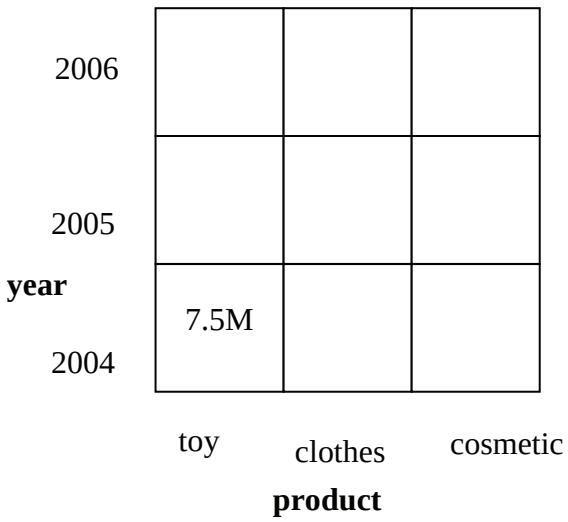


Figure 2. An example 2-D cuboid on (product, year) for the 3-D cube in Figure 1 (location=''); total sales needs to be aggregated (e.g., SUM)*



aggregate functions such as count, total, average, minimum, or maximum can be used for aggregating. Suppose the sales for toys in 2004 for NJ, NY, PA were $2.5M, $3.5M, $1.5M respectively and that the aggregating function is total. Then, the measure value for cell C2 is $7.5M.

In theory, no special operators or SQL extensions are required to take a set of records in the database and generate all the cells for the cube. Rather, the SQL group-by and union operators can be used in conjunction with dimensional sorts of the dataset to produce all cuboids. However, such an approach would be very inefficient, given the obvious interrelationships between the various group-bys produced.

MAIN THRUST

I now describe the essence of the major methods for the computation of OLAP cubes.

Top-Down Computation

In a seminal paper, Gray, Bosworth, Layman, and Pirahesh (1996) proposed the data cube operator.. The algorithm presented there forms the basis of the top-down approach.

The top-down cube computation works with non-holistic functions. An aggregate function F is called holistic if the value of F for an *n*-dimensional cell cannot be computed from a constant number of aggregates of the (*n*+1)–dimensional cell. Median and mode are examples of holistic functions, sum and avg are examples of non-holistic functions.The non-holistic functions have the property that more detailed aggregates (i.e., more dimensions) can be used to compute less detailed aggregates. This property induces a partial-ordering (i.e., a *lattice*) on all the group-bys of the cube. A group-by is called a child of some parent group-by if the parent can be used to compute the child (and no intermediate group-bys exist between the parent and child). Figure 3 depicts a sample lattice where A, B, C, and D are dimensions, nodes represent group-bys, and the edges show the parent-child relationship.

The basic idea for top-down cube construction is to start by computing the base cuboid (group-by for which no cube dimensions are aggregated). A single pass is made over the data, a record is examined, and the appropriate base cell is incremented. The remaining group-bys are computed by aggregating over already computed finer grade group-by. If a group-by can be computed from one or more possible parent group-bys, then the algorithm uses the parent smallest in size. For example, for computing the cube ABCD, the algorithm starts out by computing the base cuboid for ABCD. Then, using ABCD, it computes the cuboids for ABC, ABD, and BCD. The algorithm then repeats itself by computing the 2-D cuboids, AB, BC, AD, and

Figure 3. Cube lattice

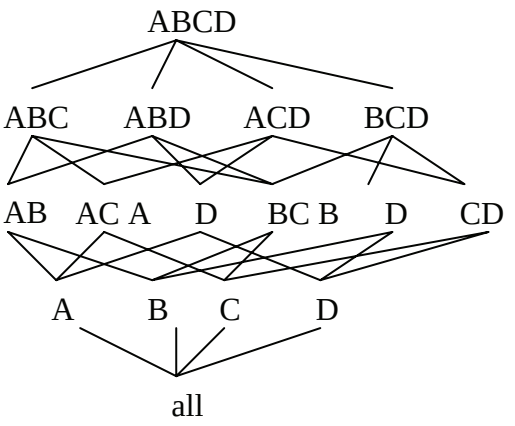
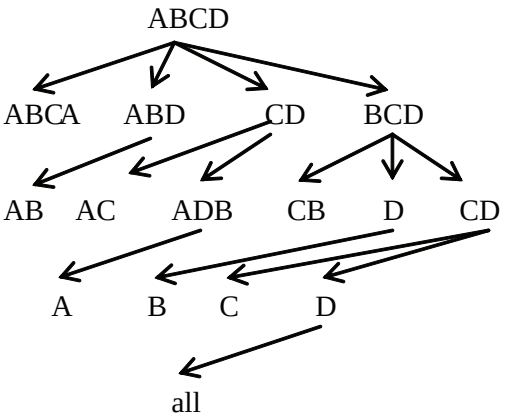


Figure 4. Top-down cube computation



BD. Note that the 2-D–cuboids can be computed from multiple parents. For example, AB can be computed from ABC or ABD. The algorithm selects the smaller group-by (the group-by with the fewest number of cells). An example top-down cube computation is shown in Figure 4.

Variants of this approach optimize on additional costs. The well-known methods are the PipeSort and PipeHash (Agarwal, Agrawal, Deshpande, Gupta, Naughton, Ramakrishnan, et al., 1996). The basic idea in the work is to extend the sort-based and hash-based methods for computing group-bys to multiple group-bys from the lattice. The optimiziations they incorporate include: computing a group-by from the smallest previously computed-group-by, reducing disk I/O by caching group-bys, reducing disk reads by computing multiple group-bys from a single parent group-by, and sharing of sorting and hashing costs across multiple group-bys.

Both PipeSort and PipeHash are examples of ROLAP (Relational-OLAP) algorithms as they operate on multidimensional tables. An alternative is MOLAP (Multidimensional OLAP), where the approach is to store the cube directly as a multidimensional array. If the data are in relational form, then they are loaded into arrays, and the cube is computed from the arrays. The advantage is that no tuples are needed for comparison purposes; all operations are performed by using array indices.

A good example for a MOLAP-based cubing algorithm is the MultiWay algorithm (Zhao, Deshpande, & Naughton, 1997). Here, to make efficient use of memory, a single large *d*-dimensional array is divided into smaller *d*-dimensional arrays, called *chunks.* It is not necessary to keep all chunks in memory at the same time; only part of the group-by arrays are needed for a given instance. The algorithm starts at the base cuboid with a single chunk. The results are passed to immediate lower-level cuboids of the top-down cube computation tree allowing computation for multiple cuboids concurrently. After the chunk processing is complete, the next chunk is loaded. The process then repeats itself. An advantage of the MultiWay algorithm is that it also allows for shared computation, where intermediate aggregate values are reused for the computation of successive descendant cuboids. The disadvantage of MOLAP-based algorithms is that they are not scalable for a large number of dimensions. because the intermediate results become too large to fit in memory. (Xin, Han, Li, & Wah, 2003).

Bottom-Up Computation

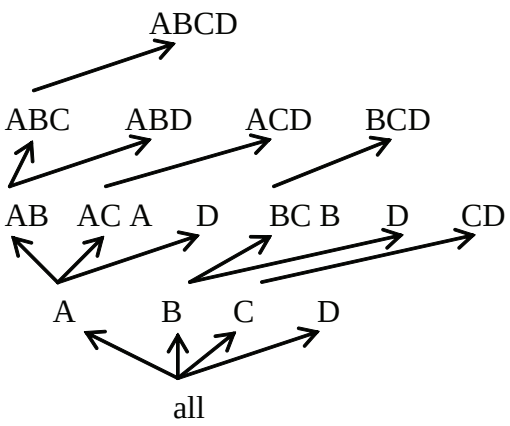
Computing and storing full data cubes may be overkill. Consider, for example, a dataset with 10 dimensions, each with a domain size of 9 (i.e., can have nine distinct values). Computing the full cube for this dataset would require computing and storing 10^{10} cuboids. Not all cuboids in the full cube may be of interest. For example, if the measure is total sales, and a user is interested in finding the cuboids that have high sales (greater than a given threshold), then computing cells with low total sales would be redundant. It would be more efficient here to only compute the cells that have total sales greater than the threshold. The group of cells that satisfies the given query is referred to as an *iceberg*

cube. The query returning the group of cells is referred to as an *iceberg query.*

Iceberg queries are typically computed using a *bottom-up* approach such as BUC (Beyer & Ramakrishnan, 1999). These methods work from the bottom of the cube lattice and then work their way up towards the cells with a larger number of dimensions. Figure 5 illustrates this approach. The approach is typically more efficient for high-dimensional, sparse datasets. The key observation to its success is in its ability to prune cells in the lattice that do not lead to useful answers. Suppose, for example, that the iceberg query is COUNT()>100. Also, suppose there is a cell X(A=a2) with COUNT=75. Then, clearly, C does not satisfy the query. However, in addition, you can see that any cell X' in the cube lattice that includes the dimensions (e.g. A=a2, B=b3) will also not satisfy the query. Thus, you need not look at any such cell after having looked at cell X. This forms the basis of support pruning. The observation was first made in the context of frequent sets for the generation of association rules (Agrawal, Imielinski, & Swami, 1993). For more arbitrary queries, detecting whether a cell is prunable is more difficult. However, some progress has been made in that front too. For example in Imielinski, Khachiyan, & Abdulghani, 2002 a method has been described for pruning arbitrary queries involving aggregates SUM, MIN, MAX, AVG and count functions.

Another feature of BUC-like algorithm is its ability to share costs during construction of different nodes. For example in Figure 5, BUC scans through the the records in a relation R and computes the aggregate cuboid node "ALL". It then sorts through R according to attribute A and finds the set of tuples for the cell X(A=a1), that share the same value a1 in A. It aggregates the measures of the tuples in this set and moves on to node AB. The input to this node are the set of tuples for the cell X(A=a1). Here, it sorts the input based on attribute B, and finds the set of tuples for the cell X(A=a1, B=b1) that share the same value b1 in B. It aggregates the measures of the tuples in this set and moves on to node ABC(assuming no pruning was performed). The input would be the set of tuples from the previous node. In node ABC, a new sort occurs on the tuples in the input set according to attribute C and set of tuples for cells X(A=a1, B=b1, C=c1), X(A=a1, B=b1, C=c2)…..X(A=a1,B=b1,C=cn) are found and aggregated. Once the node ABC is fully processed for the set of tuples in the cell X(A=a1, B=b1), the algorithm moves to the next set of tuples to process in node AB namely, the set of tuples for the cell X(A=a1, B=b2). And so on. Note, the computation occurs in a depth-first manner, and each node at a higher level uses the output of the node at the lower level. The work described in Morfonios, k. & Ioannidis, Y (2006), further generalizes this execution plan to dimensions with multiple hierarchies.

Figure 5. Bottom-up cube computation



Integrated Top-Down and Bottom-Up Approach

Typically, for low-dimension, low-cardinality, dense datasets, the top-down approach is more applicable than the bottom-up one. However, combining the two approaches leads to an even more efficient algorithm for cube computation (Xin, Han, Li, & Wah, 2003). On the global computation order, the work presented uses the top-down approach. At a sublayer underneath, it exploits the potential of the bottom-up model. Consider the top-down computation tree in Figure 4. Notice that the dimension ABC is included for all the cuboids in the leftmost subtree (the node labeled ABC).. Similarly, all the cuboids in the second and third left subtrees include the dimensions AB and A respectively. These common dimensions are termed the *shared dimensions* of the particular subtrees and enable bottom-up computation. The observation is that if a query is prunable on the cell defined by the shared dimensions, then the rest of the cells generated from this shared dimension are unneeded. The critical requirement is that for every cell X, the cell for the *shared dimensions* must be computed

first. The advantage of such an approach is that it allows for shared computation as in the top-down approach as well as for pruning, when possible.

The common theme amongst most of the algorithms discussed in literature for cube computation has been that the lattice space for computation are dimension-based. Shao, Han & Xin(2004) describes a different approach where the data density of the different dimensions in the data set are also considered in the computation order. The work factorizes the lattice space into one dense subspace and several sparse subspaces and each such subspace is handled separately. Their conclusions indicate that their approach provide efficient computation for both sparse and dense data sets.

Other Approaches

Until now, I have considered computing the cuboids from the base data. Another commonly used approach is to *materialize* the results of a selected set of group-bys and evaluate all queries by using the materialized results. Harinarayan, Rajaraman, and Ullman (1996) describe an approach to materialize a limit of k group-bys. The first group-by to materialize always includes the top group-by, as none of the group-bys can be used to answer queries for this group-by. The next group-by to materialize is included such that the *benefit* of including it in the set exceeds the benefit of any other nonmaterialized group-by. Let S be the set of materialized group-bys. This *benefit* of including a group-by v in the set S is the total savings achieved for computing the group-bys not included in S by using v versus the cost of computing them through some group-by already in S. Gupta, Harinarayan, Rajaraman, and Ullman (1997) further extend this work to include indices in the cost.

The subset of the cuboids selected for materialization is referred to as a *partial cube.* After a set of cuboids has been materialized, queries are evaluated by using the materialized results. Park, Kim, and Lee (2001) describe a typical approach. Here, the OLAP queries are answered in a three-step process. In the first step, it selects the materialized results that will be used for rewriting and identifies the part of the query (region) that the materialized result can answer. Next, query blocks are generated for these query regions. Finally, query blocks are integrated into a rewritten query.

Future Trends

In this paper, I focus on the basic aspects of cube computation. The field has matured in the last decade and a half. Some of the issues that I believe will get more attention in future work include:

- Making use of inherent property of the dataset to reduce computation of the data cubes. An example is the *range cubing algorithm,* which utilizes the correlation in the datasets to reduce the computation cost (Feng, Agrawal, Abbadi, & Metwally, 2004).
- Further work on compressing the size of the data cube and storing it efficiently. There is a realization that a great portion of the data stored in a cube is redundant and both storage and computation savings can be achieved by removing the redundancies. Examples of causes for the redundancy are repeating dimension values and repeating aggregation values across cells of the cube. Related to this issue, is also the storage format of the non-redundant data. For example, it is not very efficient storage wise if a ROLAP-based approach, stores the complete cube structure as a single relation with fixed-size tuples. Attempts to handle these issues have been described in Sismanis, Deligiannakis, Roussopoulus, and Kotidis (2002) and Morfonios, k. & Ioannidis, Y (2006) and Lakshmanan, L.V.S., Pei, J., & Zhao, Y. (2003).
- Support for indexing for answering certain cube queries fast.
- Computing cubes for richer data sets like XML–data (Wiwatwattana, N., Jagadish, H.V., Lakshmanan, L.V.S., & Srivastava, 2007).

CONCLUSION

This paper focuses on the methods for OLAP cube computation. Initial approaches shared the similarity that they make use of the ordering defined by the cube lattice to drive the computation. For the top-down approach, traversal occurs from the top of the lattice. This has the advantage that for the computation of successive descendant cuboids, intermediate node results are used. The bottom-up approach traverses the lattice in the reverse direction. The method can no longer rely on making

use of the intermediate node results. Its advantage lies in the ability to prune cuboids in the lattice that do not lead to useful answers. The integrated approach uses the combination of both methods, taking advantages of both. New alternatives have also been proposed that also take data density into consideration.

The other major option for computing the cubes is to materialize only a subset of the cuboids and to evaluate queries by using this set. The advantage lies in storage costs, but additional issues, such as identification of the cuboids to materialize, algorithms for materializing these, and query rewrite algorithms, are raised.

REFERENCES

Agarwal, S., Agrawal, R., Deshpande, P., Gupta, A., Naughton, J. F., Ramakrishnan, R., et al. (1996). On the computation of multidimensional aggregates. *Proceedings of the International Conference on Very Large Data Bases,* 506–521.

Agrawal, R., Imielinski, T., & Swami, A. N. (1993). Mining association rules between sets of items in large databases. *Proceedings of the ACM SIGMOD Conference*, 207–216.

Beyer, K. S., & Ramakrishnan, R. (1999). Bottom-up computation of sparse and iceberg cubes**.** *Proceedings of the ACM SIGMOD Conference,* 359–370.

Feng, Y., Agrawal, D., Abbadi, A. E., & Metwally, A. (2004). Range CUBE: Efficient cube computation by exploiting data correlation. *Proceedings of the International Conference on Data Engineering,* 658–670.

Gray, J., Bosworth, A., Layman, A., & Pirahesh, H. (1996). Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-total. *Proceedings of the International Conference on Data Engineering,* 152–159.

Gupta, H., Harinarayan, V., Rajaraman, A., & Ullman, J. D. (1997). Index selection for OLAP. *Proceedings of the International Conference on Data Engineering,* , 208–219.

Han, J. & Kamber,M. (2006). Data Mining: Concepts and Techniques. Morgan Kaufman Publishers. ISBN 1-55860-901-6.

Han, J., Pei, J., Dong, G., & Wang, K. (2001). Efficient computation of iceberg cubes with complex measures. *Proceedings of the ACM SIGMOD Conference,* 1–12.

Harinarayan, V., Rajaraman, A., & Ullman, J. D. (1996). Implementing data cubes efficiently. *Proceedings of the ACM SIGMOD Conference,* 205–216.

Imielinski, T., Khachiyan, L., & Abdulghani, A. (2002). Cubegrades: Generalizing association rules. *Journal of Data Mining and Knowledge Discovery, 6*(3), 219–257.

Lakshmanan, L.V.S., Pei, J., & Zhao, Y. (2003). QC-Trees: An Efficient Summary Structure for Semantic OLAP. *Proceedings of the ACM SIGMOD Conference,* 64–75.

Morfonios, k. & Ioannidis, Y (2006). CURE for Cubes: Cubing Using a ROLAP Engine. *Proceedings of the International Conference on Very Large Data Bases,* 379-390.

Park, C.-S., Kim, M. H., & Lee, Y.-J. (2001). Rewriting OLAP queries using materialized views and dimension hierarchies in data warehouses. *Proceedings of the International Conference on Data Engineering,* , 515-523.

Shao, Z., Han, J., & Xin D. (2004). MM-Cubing: Computing Iceberg Cubes by Factorizing the Lattice Space, Proceeding of the International Confernece on Scientific and Statistical Database Management, 213-222.

Sismanis, Y., Deligiannakis, A., Roussopoulus, N., & Kotidis, Y. (2002). Dwarf: Shrinking the petacube. *Proceedings of the ACM SIGMOD Conference, ,* 464–475.

Wiwatwattana, N., Jagadish, H.V., Lakshmanan, L.V.S., & Srivastava, D. (2007). X^3: A Cube Operator for XML OLAP, *Proceedings of the International Conference on Data Engineering, 916-925.*

Xin, D., Han, J., Li, X., & Wah, B. W. (2003). Star-cubing: Computing iceberg cubes by top-down and bottom-up integration**.** *Proceedings of the International Conference on Very Large Data Bases,* 476–487.

Zhao, Y., Deshpande, P. M., & Naughton, J. F. (1997). An array-based algorithm for simultaneous multidimensional aggregates. *Proceedings of the ACM SIGMOD Conference, ,* 159–170.

KEY TERMS

Bottom-Up Cube Computation: Cube construction that starts by computing from the bottom of the cube lattice and then working up toward the cells with a greater number of dimensions.

Cube Cell: Represents an association of a measure *m* with a member of every dimension.

Cuboid: A group-by of a subset of dimensions, obtained by aggregating all tuples on these dimensions.

Holistic Function: An aggregate function F is called holistic if the value of F for an *n*-dimensional cell cannot be computed from a constant number of aggregates of the (*n*+1)–dimensional cell'

Iceberg Cubes: Set of cells in a cube that satisfies an iceberg query.

Iceberg Query: A query on top of a cube that asks for aggregates above a certain threshold.

N-Dimensional Data Cube: A union of all of *k*-dimensional (k<=n) cuboids arranged by the n dimensions of the data.

Partial Cube: The subset of the cuboids selected for materialization.

Sparse Cube: A cube is sparse if a high ratio of the cube's possible cells does not contain any measure value.

Top-Down Cube Computation: Cube construction that starts by computing the base cuboid and then iteratively computing the remaining cells by aggregating over already computed finer-grade cells in the lattice.

Conceptual Modeling for Data Warehouse and OLAP Applications

Elzbieta Malinowski
Universidad de Costa Rica, Costa Rica

Esteban Zimányi
Université Libre de Bruxelles, Belgium

INTRODUCTION

The advantages of using conceptual models for database design are well known. In particular, they facilitate the communication between users and designers since they do not require the knowledge of specific features of the underlying implementation platform. Further, schemas developed using conceptual models can be mapped to different logical models, such as the relational, object-relational, or object-oriented models, thus simplifying technological changes. Finally, the logical model is translated into a physical one according to the underlying implementation platform.

Nevertheless, the domain of conceptual modeling for data warehouse applications is still at a research stage. The current state of affairs is that logical models are used for designing data warehouses, i.e., using *star* and *snowflake* schemas in the relational model. These schemas provide a multidimensional view of data where *measures* (e.g., quantity of products sold) are analyzed from different perspectives or *dimensions* (e.g., by product) and at different levels of detail with the help of *hierarchies*. On-line analytical processing (OLAP) systems allow users to perform automatic aggregations of measures while traversing hierarchies: the roll-up operation transforms detailed measures into aggregated values (e.g., daily into monthly sales) while the drill-down operation does the contrary.

Star and snowflake schemas have several disadvantages, such as the inclusion of implementation details and the inadequacy of representing different kinds of hierarchies existing in real-world applications. In order to facilitate users to express their analysis needs, it is necessary to represent data requirements for data warehouses at the conceptual level. A conceptual multidimensional model should provide a graphical support (Rizzi, 2007) and allow representing facts, measures, dimensions, and different kinds of hierarchies.

BACKGROUND

Star and snowflake schemas comprise relational tables termed *fact* and *dimension tables*. An example of star schema is given in Figure 1.

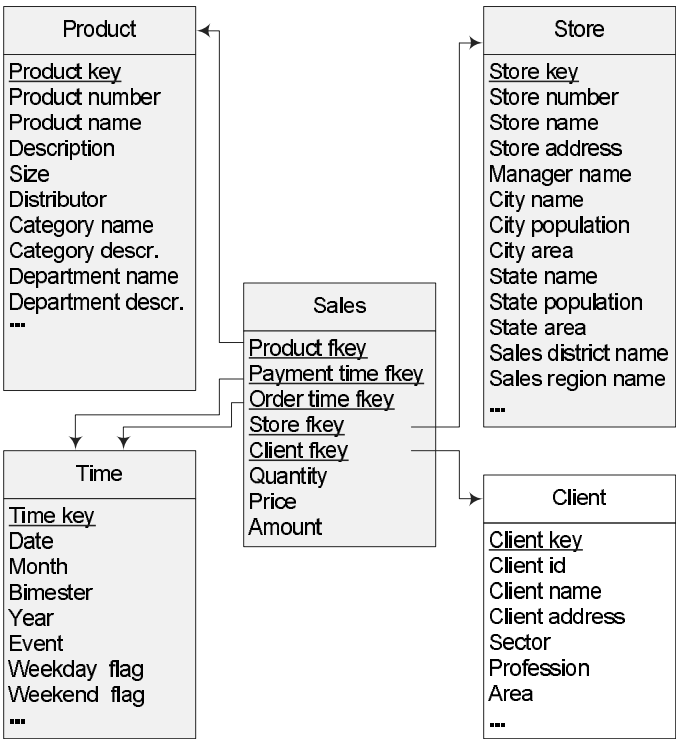
Fact tables, e.g., Sales in Figure 1, represent the focus of analysis, e.g., analysis of sales. They usually contain numeric data called *measures* representing the indicators being analyzed, e.g., Quantity, Price, and Amount in the figure. *Dimensions*, e.g., Time, Product, Store, and Client in Figure 1, are used for exploring the measures from different analysis perspectives. They often include attributes that form *hierarchies*, e.g., Product, Category, and Department in the Product dimension, and may also have descriptive attributes.

Star schemas have several limitations. First, since they use de-normalized tables they cannot clearly represent hierarchies: The hierarchy structure must be deduced based on knowledge from the application domain. For example, in Figure 1 is not clear whether some dimensions comprise hierarchies and if they do, what are their structures.

Second, star schemas do not distinguish different kinds of measures, i.e., additive, semi-additive, non-additive, or derived (Kimball & Ross, 2002). For example, Quantity is an additive measure since it can be summarized while traversing the hierarchies in all dimensions; Price is a non-additive measure since it cannot be meaningfully summarized across any dimension; Amount is a derived measure, i.e., calculated based on other measures. Although these measures require different handling during aggregation, they are represented in the same way.

Third, since star schemas are based on the relational model, implementation details (e.g., foreign keys) must be considered during the design process. This requires technical knowledge from users and also makes difficult the process of transforming the logical model to other models, if necessary.

Figure 1. Example of a star schema for analyzing sales



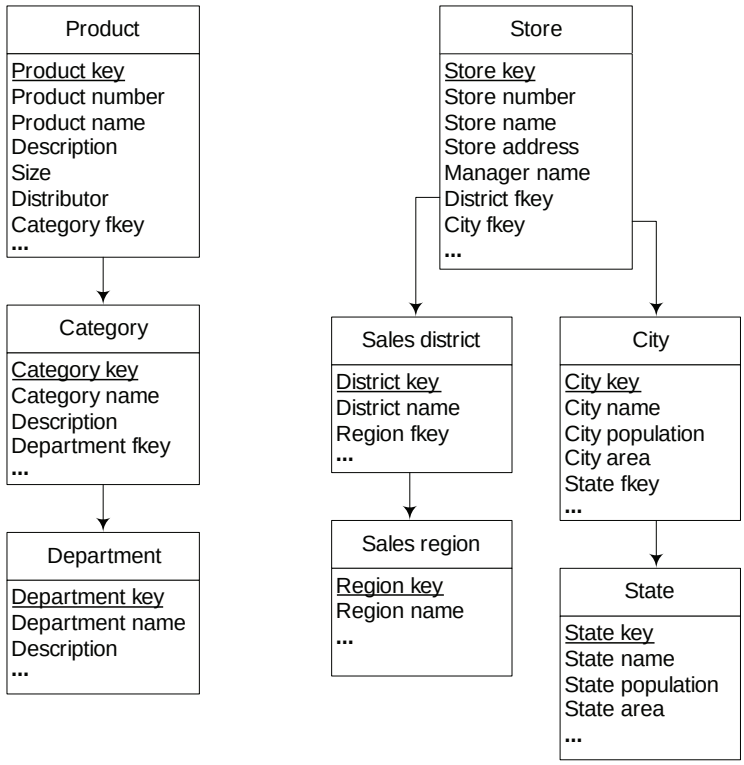
Fourth, dimensions may play different roles in a fact table. For example, the Sales table in Figure 1 is related to the Time dimension through two dates, the order date and the payment date. However, this situation is only expressed as foreign keys in the fact table that can be difficult to understand for non-expert users.

Snowflake schemas have the same problems as star schemas, with the exception that they are able to represent hierarchies. The latter are implemented as separate tables for every hierarchy level as shown in Figure 2 for the Product and Store dimensions. Nevertheless, snowflake schemas only allow representing simple hierarchies. For example, in the hierarchy in Figure 2 a) it is not clear that the same product can belong to several categories but for implementation purposes only the primary category is kept for each product. Furthermore, the hierarchy formed by the Store, Sales district, and Sales region tables does not accurately represent users' requirements: since small sales regions are not divided into sales districts, some stores must be analyzed using the hierarchy composed only of the Store and the Sales region tables.

Several conceptual multidimensional models have been proposed in the literature[1]. These models include the concepts of facts, measures, dimensions, and hierarchies. Some of the proposals provide graphical representations based on the ER model (Sapia, Blaschka, Höfling, & Dinter, 1998; Tryfona, Busborg, & Borch, 1999), on UML (Abelló, Samos, & Saltor, 2006; Luján-Mora, Trujillo, & Song, 2006), or propose new notations (Golfarelli & Rizzi, 1998; Hüsemann, Lechtenbörger, & Vossen, 2000), while other proposals do not refer to graphical representations (Hurtado & Gutierrez, 2007; Pourabbas, & Rafanelli, 2003; Pedersen, Jensen, & Dyreson, 2001; Tsois, Karayannidis, & Sellis, 2001).

Very few models distinguish the different types of measures and refer to role-playing dimensions (Kimball & Ross, 2002, Luján-Mora *et al.*, 2006). Some models do not consider the different kinds of hierarchies existing in real-world applications and only support simple hierarchies (Golfarelli & Rizzi, 1998; Sapia *et al.*, 1998). Other models define some of the hierarchies described in the next section (Abelló *et al.*, 2006; Bauer, Hümmer,

Figure 2. Snowflake schemas for the a) product and b) store dimensions from Figure 1



& Lehner, 2000; Hüsemann *et al.*, 2000; Hurtado & Gutierrez, 2007; Luján-Mora *et al.*, 2006; Pourabbas, & Rafanelli, 2003; Pedersen *et al.*, 2001; Rizzi, 2007). However, there is a lack of a general classification of hierarchies, including their characteristics at the schema and at the instance levels.

MAIN FOCUS

We present next the MultiDim model (Malinowski & Zimányi, 2004; Malinowski & Zimányi, 2008), a conceptual multidimensional model for data warehouse and OLAP applications. The model allows representing different kinds of hierarchies existing in real-world situations.

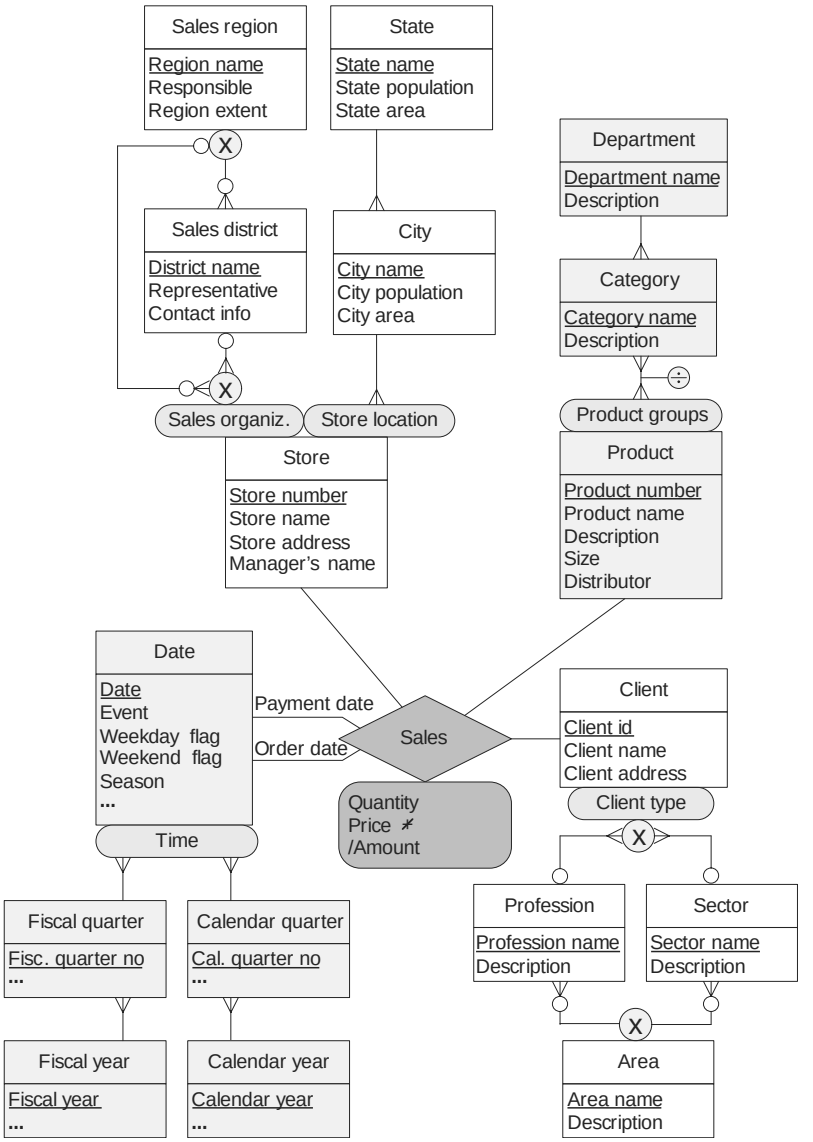
The MultiDim Model

To describe the MultiDim model we use the schema in Figure 3, which corresponds to the relational schemas in Figure 1. This schema also contains different types of hierarchies that are defined in next section.

A *schema* is composed of a set of levels organized into dimensions and a set of fact relationships. A *level* corresponds to an entity type in the ER model; instances of levels are called *members*. A level has a set of attributes describing the characteristics of their members. For example, the Product level in Figure 3 includes the attributes Product number, Product name, etc. In addition, a level has one or several keys (underlined in the figure) identifying uniquely the members of a level.

A *fact relationship* expresses the focus of analysis and represents an n-ary relationship between levels. For example, the Sales fact relationship in Figure 3 relates the Product, Date, Store, and Client levels. Further, the same level can participate several times in a fact relationship playing different roles. Each *role* is identified by a name and is represented by a separate link between the level and the fact relationship, as can be seen for the roles Payment date and Order date relating the Date level to the Sales fact relationship.

Figure 3. A conceptual multidimensional schema of a sales data warehouse



A fact relationship may contain attributes commonly called *measures*, e.g., Quantity, Price, and Amount in Figure 3. Measures are classified as *additive*, *semi-additive*, or *non-additive* (Kimball & Ross, 2002). By default we suppose that measures are additive. For semi-additive and non-additive measures we use, respectively, the symbols +! and ≠ (the latter is shown for the Price measure). For derived measures and attributes we use the symbol / in front of the measure name, as shown for the Amount measure.

A *dimension* is an abstract concept grouping data that shares a common semantic meaning within the domain being modeled. It is composed of either one level or one more hierarchies.

Hierarchies are used for establishing meaningful aggregation paths. A hierarchy comprises several related levels, e.g., the Product, Category, and Department levels. Given two related levels, the lower level is called *child*, the higher level is called *parent*, and the relationship between them is called *child-parent relationship*. Key attributes of a parent level define how

child members are grouped. For example, in Figure 3 the Department name in the Department level is used for grouping different category members during roll-up operations. A level that does not have a child level is called *leaf*; it must be the same for all hierarchies in a dimension. The leaf level name is used for defining the dimension's name. The level that does not have a parent level is called *root*.

Child-parent relationships are characterized by *cardinalities*, indicating the minimum and the maximum number of members in one level that can be related to members in another level. The notations used for representing cardinalities are as follows: —< (1,n), —o< (0,n), —— (1,1), and —o (0,1). For example, in Figure 3 the child level Store is related to the parent level City with a many-to-one cardinality, which means that every store belongs to only one city and each city can have many stores.

Since the hierarchies in a dimension may express different conceptual structures used for analysis purposes, we include an *analysis criterion* to differentiate them. For example, the Store dimension includes two hierarchies: Store location and Sales organization.

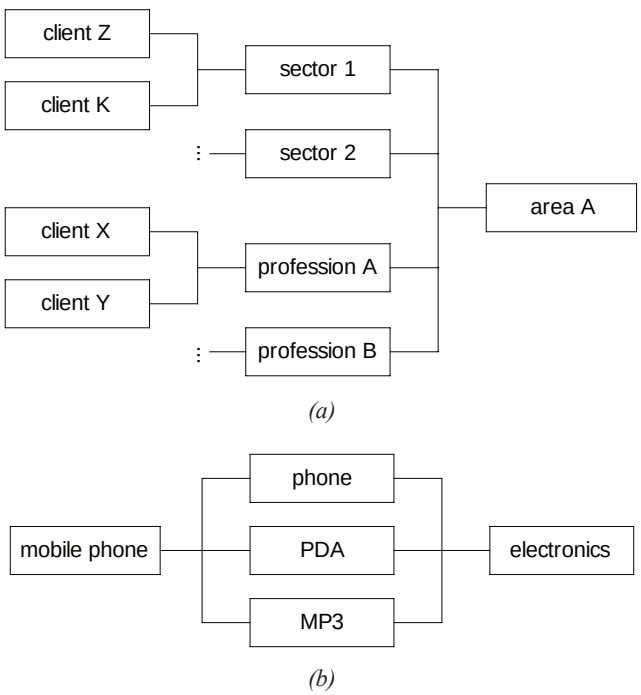
Classification of Hierarchies

We propose next a classification of hierarchies considering their differences at the schema and at the instance levels. Figure 4 shows examples of members for the Customer and Product dimensions from Figure 3. The distinction at the instance level is important since different aggregation procedures are required depending on whether the hierarchy members form a tree or an acyclic graph. This may be deduced from the cardinalities included in the schema.

We distinguish the following types of hierarchies.

- **Simple:** In these hierarchies the relationship between their members can be represented as a tree. They can be of different types.
  - **Balanced:** An example of this hierarchy is the Store location in Figure 3. At the schema level there is only one path where all levels are mandatory. At the instance level the members form a tree where all the branches have the same length. A parent member has one or several child members (at least one)

Figure 4. Examples of members for a) the Customer and b) the Product dimensions



and a child member belongs to only one parent member: the cardinality of child roles is (1,1) and that of parent roles is (1,n).

- **Unbalanced:** This type of hierarchy is not included in Figure 3; however, they are present in many data warehouse applications. For example, a bank may include a hierarchy composed by the levels ATM, Agency, and Branch. However, some agencies may not have ATMs and small branches may not have any organizational division. This kind of hierarchy has only one path at the schema level. However, since at the instance level, some parent members may not have associated child members, the cardinality of the parent role is (0,n).
- **Generalized:** The Client type hierarchy in Figure 3 with instances in Figure 4 a) belongs to this type. In this hierarchy a client can be a person or a company having in common the Client and Area levels. However, the buying behavior of clients can be analyzed according to the specific level Profession for a person type, and Sector for a company type. This kind of hierarchy has at the schema level multiple exclusive paths sharing some levels. All these paths represent one hierarchy and account for the same analysis criterion. At the instance level each member of the hierarchy only belongs to one path. We use the symbol ⊗ for indicating that the paths are exclusive. **Non-covering** hierarchies (the Sales organization hierarchy in Figure 3) are generalized hierarchies with the additional restriction that the alternative paths are obtained by skipping one or several intermediate levels. At the instance level every child member has only one parent member.

- **Non-strict:** An example is the Product groups hierarchy in Figure 3 with members in Figure 4 b). This hierarchy models the situation when mobile phones can be classified in different products categories, e.g., phone, PDA, and MP3 player. Non-strict hierarchies have at the schema level at least one many-to-many cardinality, e.g., between the Product and Category levels in Figure 3. A hierarchy is called *strict* if all cardinalities are many-to-one. Since at the instance level, a child member may have more than one parent member, the members form an acyclic graph. To indicate how the measures are distributed between several parent members, we include a distributing factor symbol ⊜. The different kinds of hierarchies previously presented can be either strict or non-strict.
- **Alternative:** The Time hierarchy in the Date dimension is a alternative hierarchy. At the schema level there are several non-exclusive simple hierarchies sharing at least the leaf level, all these hierarchies accounting for the same analysis criterion. At the instance level such hierarchies form a graph since a child member can be associated with more than one parent member belonging to different levels. In such hierarchies it is not semantically correct to simultaneously traverse the different composing hierarchies. Users must choose one of the alternative hierarchies for analysis, e.g., either the hierarchy composed by Date, Fiscal quarter, and Fiscal year or the one composed by Date, Calendar quarter, and Calendar year.
- **Parallel:** A dimension has associated several hierarchies accounting for different analysis criteria. Parallel hierarchies can be of two types. They are *independent,* if the composing hierarchies do not share levels; otherwise, they are *dependent*. The Store dimension includes two parallel independent hierarchies: Sales organization and Store location. They allow analyzing measures according to different criteria.

The schema in Figure 3 clearly indicates users' requirements concerning the focus of analysis and the aggregation levels represented by the hierarchies. It also preserves the characteristics of star or snowflake schemas providing at the same time a more abstract conceptual representation. Notice that even though the schemas in Figures 1 and 3 include the same hierarchies, they can be easily distinguished in the Figure 3 while this distinction cannot be made in Figure 1.

Existing multidimensional models do not consider all types of hierarchies described above. Some models give only a description and/or a definition of some of the hierarchies, without a graphical representation. Further, for the same type of hierarchy different names are used in different models. Strict hierarchies are included explicitly or implicitly in all proposed models.

Since none of the models takes into account different analysis criteria, alternative and parallel hierarchies cannot be distinguished. Further, very few models propose a graphical representation for the different hierarchies that facilitate their distinction at the schema and instance levels.

To show the feasibility of implementing the different types of hierarchies, we present in Malinowski and Zimányi, (2006, 2008) their mappings to the relational model and give examples of their implementation in Analysis Services and Oracle OLAP.

FUTURE TRENDS

Even though some of the presented hierarchies are considered as an advanced feature of multidimensional models (Torlone, 2003), there is a growing interest in having them in the research community and in commercial products.

Nevertheless, several issues have yet to be addressed. It is necessary to develop aggregation procedures for all types of hierarchies defined in this paper. Some proposals for managing them exist (e.g., Pedersen *et al.*, 2001; Abelló *et al.*, 2006; Hurtado & Gutierrez, 2007). However, not all hierarchies are considered and some of the proposed solutions may not be satisfactory for users since they transform complex hierarchies into simpler ones to facilitate their manipulation.

Another issue is to consider the inclusion of other ER constructs in the multidimensional model, such as weak entity types, multivalued or composite attributes. The inclusion of these features is not straightforward and requires analysis of their usefulness in multidimensional modeling.

CONCLUSION

Data warehouse and OLAP applications use a multidimensional view of data including dimensions, hierarchies, facts, and measures. In particular, hierarchies are important in order to automatically aggregate the measures for analysis purposes.

We advocated that it is necessary to represent data requirements for data warehouse and OLAP applications at a conceptual level. We proposed the MultiDim model, which includes graphical notations for the different elements of a multidimensional model. These notations allow a clear distinction of each hierarchy type taking into account their differences at the schema and at the instance levels.

We also provided a classification of hierarchies. This classification will help designers to build conceptual models of multidimensional databases. It will also give users a better understanding of the data to be analyzed, and provide a better vehicle for studying how to implement such hierarchies using current OLAP tools. Further, the proposed hierarchy classification provides OLAP tool implementers the requirements needed by business users for extending the functionality offered by current tools.

REFERENCES

Abelló, A., Samos, J., & Saltor, F. (2006). YAM^2: a multidimensional conceptual model extending UML. *Information Systems, 32*(6), pp. 541-567.

Bauer, A., Hümmer, W., & Lehner, W. (2000). An alternative relational OLAP modeling approach. *Proceedings of the 2nd International Conference on Data Warehousing and Knowledge Discovery*, pp. 189-198. Lectures Notes in Computer Sciences, N° 1874. Springer.

Golfarelli, M., & Rizzi, S. (1998). A methodological framework for data warehouse design. *Proceedings of the 1st ACM International Workshop on Data Warehousing and OLAP*, pp. 3-9.

Hurtado, C., & Gutierrez, C. (2007). Handling structural heterogeneity in OLAP. In R. Wrembel & Ch. Koncilia (Eds.), *Data Warehouses and OLAP: Concepts, Architectures and Solutions*, pp. 27-57. Idea Group Publishing.

Hüsemann, B., Lechtenbörger, J., & Vossen, G. (2000). Conceptual data warehouse design. *Proceedings of the 2nd International Workshop on Design and Management of Data Warehouses*, p. 6.

Kimball, R., & Ross, M. (2002). *The Data Warehouse Toolkit*. John Wiley & Sons Publishers.

Luján-Mora, S., Trujillo, J. & Song, I. (2006). A UML profile for multidimensional modeling in data warehouses. *Data & Knowledge Engineering, 59*(3), pp. 725-769.

Malinowski, E. & Zimányi, E. (2004). OLAP hierarchies: A conceptual perspective. *Proceedings of the 16th International Conference on Advanced Information Systems Engineering*, pp. 477-491. Lectures Notes in Computer Sciences, N° 3084. Springer.

Malinowski, E. & Zimányi, E. (2006). Hierarchies in a multidimensional model: from conceptual modeling to logical representation. *Data & Knowledge Engineering*, 59(2), pp. 348-377.

Malinowski, E. & Zimányi, E. (2008). *Advanced Data Warehouse Design: From Conventional to Spatial and Temporal Applications*. Springer.

Pedersen, T., Jensen, C.S., & Dyreson, C. (2001). A foundation for capturing and querying complex multidimensional data. *Information Systems, 26*(5), pp. 383-423.

Pourabbas, E., & Rafanelli, M. (2003). Hierarchies. In Rafanelli, M. (Ed.) *Multidimensional Databases: Problems and Solutions,* pp. 91-115. Idea Group Publishing.

Rizzi, S. (2007). Conceptual modeling solutions for the data warehouse. In R. Wrembel & Ch. Koncilia (Eds.), *Data Warehouses and OLAP: Concepts, Architectures and Solutions*, pp. 1-26. Idea Group Publishing.

Sapia, C., Blaschka, M., Höfling, G., & Dinter, B. (1998). Extending the E/R model for multidimensional paradigms. *Proceedings of the 17th International Conference on Conceptual Modeling*, pp. 105-116. Lectures Notes in Computer Sciences, N° 1552. Springer.

Torlone, R. (2003). Conceptual multidimensional models. In Rafanelli, M. (Ed.) *Multidimensional Databases: Problems and Solutions*, pp. 91-115. Idea Group Publishing.

Tryfona, N., Busborg, F., & Borch, J. (1999). StarER: A Conceptual model for data warehouse design. *Proceedings of the 2nd ACM International Workshop on Data Warehousing and OLAP*, pp. 3-8.

Tsois, A., Karayannidis, N., & Sellis, T. (2001). MAC: Conceptual data modeling for OLAP. *Proceedings of the 3rd International Workshop on Design and Management of Data Warehouses*, p. 5.

KEY TERMS

Conceptual Model: A model for representing schemas that are designed to be as close as possible to users' perception, not taking into account any implementation considerations.

MultiDim Model: A conceptual multidimensional model used for specifying data requirements for data warehouse and OLAP applications. It allows one to represent dimensions, different types of hierarchies, and facts with associated measures.

Dimension: An abstract concept for grouping data sharing a common semantic meaning within the domain being modeled.

Hierarchy: A sequence of levels required for establishing meaningful paths for roll-up and drill-down operations.

Level: A type belonging to a dimension. A level defines a set of attributes and is typically related to other levels for defining hierarchies.

Multidimensional Model: A model for representing the information requirements of analytical applications. A multidimensional model comprises facts, measures, dimensions, and hierarchies.

Star Schema: A relational schema representing multidimensional data using de-normalized relations.

Snowflake Schema: A relational schema representing multidimensional data using normalized relations.

ENDNOTE

1 A detailed description of proposals for multidimensional modeling can be found in Torlone (2003).

Constrained Data Mining

Brad Morantz
Science Applications International Corporation, USA

INTRODUCTION

Mining a large data set can be time consuming, and without constraints, the process could generate sets or rules that are invalid or redundant. Some methods, for example clustering, are effective, but can be extremely time consuming for large data sets. As the set grows in size, the processing time grows exponentially.

In other situations, without guidance via constraints, the data mining process might find morsels that have no relevance to the topic or are trivial and hence worthless. The knowledge extracted must be comprehensible to experts in the field. (Pazzani, 1997) With time-ordered data, finding things that are in reverse chronological order might produce an impossible rule. Certain actions always precede others. Some things happen together while others are mutually exclusive. Sometimes there are maximum or minimum values that can not be violated. Must the observation fit all of the requirements or just most. And how many is "most?"

Constraints attenuate the amount of output (Hipp & Guntzer, 2002). By doing a first-stage constrained mining, that is, going through the data and finding records that fulfill certain requirements before the next processing stage, time can be saved and the quality of the results improved. The second stage also might contain constraints to further refine the output. Constraints help to focus the search or mining process and attenuate the computational time. This has been empirically proven to improve cluster purity. (Wagstaff & Cardie, 2000)(Hipp & Guntzer, 2002)

The theory behind these results is that the constraints help guide the clustering, showing where to connect, and which ones to avoid. The application of user-provided knowledge, in the form of constraints, reduces the hypothesis space and can reduce the processing time and improve the learning quality.

BACKGROUND

Data mining has been defined as the process of using historical data to discover regular patterns in order to improve future decisions. (Mitchell, 1999) The goal is to extract usable knowledge from data. (Pazzani, 1997) It is sometimes called knowledge discovery from databases (KDD), machine learning, or advanced data analysis. (Mitchell, 1999)

Due to improvements in technology, the amount of data collected has grown substantially. The quantities are so large that proper mining of a database can be extremely time consuming, if not impossible, or it can generate poor quality answers or muddy or meaningless patterns. Without some guidance, it is similar to the example of a monkey on a typewriter: Every now and then, a real word is created, but the vast majority of the results is totally worthless. Some things just happen at the same time, yet there exists no theory to correlate the two, as in the proverbial case of skirt length and stock prices.

Some of the methods of deriving knowledge from a set of examples are: association rules, decision trees, inductive logic programming, ratio rules, and clustering, as well as the standard statistical procedures. Some also use neural networks for pattern recognition or genetic algorithms (evolutionary computing). Semi-supervised learning, a similar field, combines supervised learning with self-organizing or unsupervised training to gain knowledge (Zhu, 2006) (Chappelle et al., 2006). The similarity is that both constrained data mining and semi-supervised learning utilize the a-priori knowledge to help the overall learning process.

Unsupervised and unrestricted mining can present problems. Most clustering, rule generation, and decision tree methods have order O much greater than N, so as the amount of data increases, the time required to generate clusters increases at an even faster rate. Additionally, the size of the clusters could increase, making it harder to find valuable patterns. Without

constraints, the clustering might generate rules or patterns that have no significance or correlation to the problem at hand. As the number of attributes grows, the complexity and the number of patterns, rules, or clusters grows exponentially, becoming unmanageable and overly complex. (Perng et al,2002)

A constraint is a restriction; a limitation. By adding constraints, one guides the search and limits the results by applying boundaries of acceptability. This is done when retrieving the data to search (i.e. using SQL) and/or during the data mining process. The former reduces the amount of data that will be organized and processed in the mining by removing extraneous and unacceptable regions. The latter is what directly focuses the process to the desired results.

MAIN FOCUS

Constrained Data Mining Applications

Constrained data mining has been said to be the "best division of labor," where the computer does the number crunching and the human provides the focus of attention and direction of the search by providing search constraints. (Han et al, 1999) Constraints do two things: 1) They limit where the algorithm can look; and 2) they give hints about where to look. (Davidson & Ravi, 2005) As a constraint is a guide to direct the search, combining knowledge with inductive logic programming is a type of constraint, and that knowledge directs the search and limits the results. This combination is extremely effective. (Muggleton, 1999)

If every possible pattern is selected and the constraints tested afterwards, then the search space becomes large and the time required to perform this becomes excessive. (Boulicaut & Jeudy, 2005) The constraints must be in place during the search. They can be as simple as thresholds on rule quality measure support or confidence, or more complicated logic to formulate various conditions. (Hipp & Guntzer, 2002)

In mining with a structured query language (SQL), the constraint can be a predicate for association rules. (Ng et al, 1998) In this case, the rule has a constraint limiting which records to select. This can either be the total job or produce data for a next stage of refinement. For example, in a large database of bank transactions, one could specify only records of ACH transactions that occurred during the first half of this year. This reduces the search space for the next process.

A typical search would be:

```
select * where year = 2006 and where month < 7
```

It might be necessary that two certain types always cluster together (must-link), or the opposite, that they may never be in the same cluster (cannot-link). (Ravi & Davidson, 2005) In clustering (except fuzzy clustering), elements either are or are not in the same cluster. (Boulicaut & Jeudy, 2005) Application of this to the above example could further require that the transactions must have occurred on the first business day of the week (must-link), even further attenuating the dataset. It could be even further restricted by adding a cannot-link rule such as not including a national holiday. In the U.S.A., this rule would reduce the search space by a little over 10 percent. The rule would be similar to:

```
select * where day = monday and day <8 and where
day \=holiday
```

If mining with a decision tree, pruning is an effective way of applying constraints. This has the effect of pruning the clustering dendogram (clustering tree). If none of the elements on the branch meet the constraints, then the entire branch can be pruned. (Boulicaut & Jeudy, 2005) In Ravi and Davidson's study of image location for a robot, the savings from pruning were between 50 percent and 80 percent. There was also a typical improvement of a 15 percent reduction in distortion in the clusters, and the class label purity improved. Applying this to the banking example, any branch that had a Monday national holiday could be deleted. This would save about five weeks a year, or about 10 percent.

The Constraints

Types of constraints:

1. **Knowledge-based:** what type of relationships are desired, association between records, classification, prediction, or unusual repetitive patterns
2. **Data-based:** range of values, dates or times, relative values
3. **Rules:** time order, relationships, acceptable patterns

4. **Statistical:** at what levels are the results interesting (Han et al, 1999)
5. **Linking:** must-link and cannot-link (Davidson & Ravi, 2005)

Examples of constraints:

1. Borders
2. Relationships
3. Syntax
4. Chronology
5. Magnitude
6. Phase
7. Frequency

Sources of constraints:

1. A priori knowledge
2. Goals
3. Previous mining of the data set
4. The analyst
5. Insight into the data, problem, or situation
6. Time
7. User needs
8. Customer information or instruction
9. Domain knowledge

Attribute relationships and dependencies can provide needed constraints. Expert and/or domain knowledge, along with user preferences, can provide additional rules. (Perng et al., 2002)

Common or domain knowledge as well as item taxonomies can add insight and direct which attributes should be included or excluded, how they should be used in the grouping, and the relationships between the two. (Perng et al., 2002)

When observations occur over time, some data occurs at a time when it could not be clustered with another item. For example, if A always occurs before B, then one should only look at records where this is the case. Furthermore, there can also be a maximum time between the two observations; if too much time occurs between A and B, then they are not related. This is a mathematical relationship, setting a window of allowable time. The max-gap strategy sets a limit for the maximum time that can be between two elements. (Boulicaut & Jeudy, 2005) An example is:

IF ((time(A) < time(B)) AND (time(B) – time(A) < limit))

Domain knowledge can also cause groupings. Some examples are: knowing that only one gender uses a certain item, most cars have four tires, or that something is only sold in pairs. These domain knowledge rules help to further divide the search space. They produce limitations on the data.

Another source of constraints is the threshold of how many times or what percentage the discovered pattern has occurred in the data set. (Wojciechowski & Zakrewicz, 2002) Confidence and support can create statistical rules that reduce the search space and apply probabilities to found rules. Using ratio rules is a further refinement to this approach, as it predicts what will be found based upon past experience. (Korn et al., 2000)

Using the Constraints

User interactive constraints can be implemented by using a data mining query language generating a class of conditions that will extract a subset of the original database. (Goethals & van der Bussche, 2003) Some miners push the constraints into the mining process to attenuate the number of candidates at each level of the search. (Perng et al., 2002) This might suffice or further processing might be done to refine the data down to the desired results. Our research has shown that if properly done, constraints in the initial phase not only reduce computation time, but also produce clearer and more concise rules, with clusters that are statistically well defined. Setting the constraints also reduces computation time in clustering by reducing the number of distance calculations required. Using constraints in clustering increases accuracy while it reduces the number of iterations, which translates to faster processing times. (Davidson & Ravi, 2005)

Some researchers claim that sometimes it is more effective to apply the constraints after the querying process, rather than within the initial mining phase. (Goethals & van der Bussche, 2003) Using constraints too early in the process can have shortcomings or loss of information, and therefore, some recommend applying constraints later. (Hipp & Guntzer, 2002) The theory is that something might be excluded that should not have been.

Ratio rules (Korn et al, 2000) allow sorting the data into ranges of ratios, learning ratio patterns, or finding anomalies and outliers. Ratio rules allow a better understanding of the relationships within the data. These ratios are naturally occurring, in everyday life. A person wears, one skirt, one blouse, and zero or one hat. The numbers do not vary from that. In many things in life, ratios exist, and employing them in data mining will improve the results.

Ratio rules offer the analyst many opportunities to craft additional constraints that will improve the mining process. These rules can be used to look at anomalies and see why they are different, to eliminate the anomalies and only consider average or expected performances, split the database into ratio ranges, or only look within certain specified ranges. Generating ratio rules requires a variance-covariance matrix, which can be time- and resource-consuming.

In a time-ordered database, it is possible to relate the changes and put them into chronological order. The changes then become the first derivative over time and can present new information. This new database is typically smaller than the original one. Mining the changes database can reveal even more information.

In looking for relationships, one can look in both directions, both to see A occur and then look for B, or after finding B, look for the A that could have been the cause. After finding an A, the search is now only for the mating B, and nothing else, reducing even more the search space and processing time. A maximum time interval makes this a more powerful search criteria. This further reduces the search space.

User desires, a guide from the user, person, or group commissioning the data mining, is an important source. Why spend time and energy getting something that is not wanted. This kind of constraint is often overlooked but is very important. The project description will indicate which items or relationships are not of interest, are not novel, or are not of concern. The user, with this domain knowledge, can direct the search.

Domain knowledge, with possible linking to other databases, will allow formulating a maximum or minimum value of interrelation that can serve as a limit and indicate a possibility. For example, if a person takes the bus home, and we have in our database the bus schedule, then we have some expected times and limits for the arrival time at home. Furthermore, if we knew how far the person lived from the bus stop, and how fast the person walked, we could calculate a window of acceptable times.

Calculations can be done on the data to generate the level of constraint. Some of this can be done in the query, and more complex mathematics can be implemented in post processing. Our research has used IDL for exploratory work and then implemented in Fortran 95 for automated processing. Fortran 95, with its matrix manipulations and built-in array functions, allowed fast programming. The question is not which constraints to use, but rather what is available. In applying what knowledge is in hand, the process is more focused and produces more useful information.

FUTURE TRENDS

An automated way to implement the constraints simply into the system without having to hard code them into the actual algorithm (as the author did in his research) would simplify the process and facilitate acceptance into the data mining community. Maybe a table where rules can be entered, or possibly a specialized programming language, would facilitate this process. The closer that it can approach natural language, the more likely it is to be accepted. The entire process is in its infancy and, over time, will increase performance both in speed and accuracy. One of the most important aspects will be the ability to uncover the hidden nuggets of information or relatioships that are both important and unexpected.

With the advent of multi-core processors, the algorithms for searching and clustering the data will change in a way to fully (or at least somewhat) utilize this increased computing power. More calculations and comparisons will be done on the fly as the data is being processed. This increased computing power will not only process faster, but will allow for more intelligent and computationally intensive constraints and directed searches.

CONCLUSION

Large attenuation in processing times has been proven by applying constraints that have been derived from user or domain knowledge, attribute relationships and dependencies, customer goals, and insights into given problems or situations (Perng et al., 2002). In applying these rules, the quality of the results improves, and the results are quite often more meaningful.

REFERENCES

Boulicaut, J-P., & Jeudy, B., (2005). Constraint-Based Data Mining, *The Data Mining and Knowledge Discovery Handbook 2005*, Maimon, O. & Rokach, L. Eds Springer-Verlag 399-416.

Chappelle,O., Scholkopf, B., & Zein, A. (2006). *Semi Supervised Learning*, MIT Press, Cambridge MA.

Davidson, I., & Ravi, S. (2005). Agglomerative Hierarchical Clustering with Constraints: Theoretical and Empirical Results, *Proceedings of PKDD 2005 9th European Conference on Principles and Practice of Knowledge Discovery in Database,* Jorge, A., P., & Gama, J, Eds. Springer, 59-70.

Davidson, I., & Ravi, S. (2005). Clustering Under Constraints: Feasibility Issues and the k-Means Algorithm, *Papers Presented in the 2005 SIAM International Data Mining Conference.*

Goethals, B., & van der Bussche, J. (2000). On Supporting Interactive Constrained Association Rule Mining, *Proceedings of the Second International Conference on Data Warehousing and Knowledge Discovery*; Vol. 1874, pages 307–316, Springer-Verlag.

Han, J., Lakshmanan, L., & Ng, R. (1999). Constraint-Based Multidimensional Data Mining, *IEEE Computer*, 32(8), August 1999.

Hipp, J., & Guntzer, U. (). Is Pushing Constraints Deeply Into the Mining Algorithm Really What We Want? An Alternative Approach to Association Rule Mining, *SIGKDD*, 4(1), 50-55.

Korn, F., Kotidis, Y., Faloutsos, C., & Labrinidis, A., (2000). Quantifiable Data Mining Using Ratio Rules, *The International Journal of Very Large Data Bases*, 8(3-4), 254-266.

Mitchell, T. (1999). Machine Learning and Data Mining, *Communications of the ACM*, 42(11), November 1999, 30-36.

Muggleton, S. (1999). Scientific Knowledge Discovery Using Inductive Logic Programming, *Communications of the ACM*, 42(11), November 1999, 42-46.

Ng, R., Laks, V., Lakshmanan, S., Han, J., & Pang, A. (1998) Exploratory Mining and Pruning Optimizations of Constrained Association Rules, *Proceedings of ACM SIGMOD International Conference on Management of Data,* Haas, L, & Tiwary, A. Eds., ACM Press, 13-24.

Pazzani, M., Mani, S.,& Shankle, W. (1997). Comprehensible Knowledge-Discovery in Databases, *Proceedings from the 1997 Cognitive Science Society Conference, 596-601.* Lawrence Erlbaum.

Perng, C., Wang, H., Ma, S., & Hellerstein, J. (2002). Discovery in Multi-Attribute Data with User-Defined Constraints, *SIGKDD Explorations,* 4(1), 56-64.

Srikant, R. & Vu, Q. (1997). Mining Association Rules with Item Constraints, *Proceedings of the 3rd International Conference of Knowledge Discovery And Data Mining*, Heckerman, D., Mannila, H., Pregibon, D., & Uthurusamy, R. Eds, AAAI Press, 67-73.

Wagstaff, K., & Cardie, C. (2000). Clustering with Instance Level Constraints, *ICML*, in Davidson & Ravi

Wojciechowski, M., & Zakrzewicz, M, (2002). Dataset Filtering Techniques in Constraint-Based Frequent Pattern Mining. *Proceedings of the ESF Exploratory Workshop on Pattern Detection and Discovery, ACM Lecture Notes in Computer Science* 2447, 77-91.

Zhu, X. (2006). *Semi-supervised Learning Literature Survey*, doctoral thesis, http://www.cs.wisc.edu/~jerryzhu/research/ssl/semireview.html

KEY TERMS

Anomaly: An irregular value, or one that is beyond the limits of acceptability; in statistics, a value further from the mean than a specified distance, typically 2 or 3 sigma.

Association Rule: A data mining term for a rule that describes items in a basket that are dependent on other items also in the basket.

Cluster: A collection of like items, similar along specified dimensions. Cluster analysis is a process where items or data points are sorted by their similarities and separated by their differences.

Constraint: A restriction, a forced guideline, a specific confining rule; in data mining, it is an exacting, specific rule defining specific intents and areas.

Fortran 95: A mathematical programming language, one of the first computer languages ever developed; the latest versions are optimized for matrix operations and multiprocessing.

Ratio Rule: A data mining term for a quantity relationship between attributes.

Tree building: A mathematical process if generating a decision tree that is data-driven and can be used for classification.

Constraint-Based Association Rule Mining

Carson Kai-Sang Leung
The University of Manitoba, Canada

INTRODUCTION

The problem of association rule mining was introduced in 1993 (Agrawal et al., 1993). Since then, it has been the subject of numerous studies. Most of these studies focused on either performance issues or functionality issues. The former considered *how* to compute association rules efficiently, whereas the latter considered *what* kinds of rules to compute. Examples of the former include the Apriori-based mining framework (Agrawal & Srikant, 1994), its performance enhancements (Park et al., 1997; Leung et al., 2002), and the tree-based mining framework (Han et al., 2000); examples of the latter include extensions of the initial notion of association rules to other rules such as dependence rules (Silverstein et al., 1998) and ratio rules (Korn et al., 1998). In general, most of these studies basically considered the data mining exercise in isolation. They did not explore how data mining can interact with the human user, which is a key component in the broader picture of knowledge discovery in databases. Hence, they provided little or no support for user focus. Consequently, the user usually needs to wait for a long period of time to get numerous association rules, out of which only a small fraction may be interesting to the user. In other words, the user often incurs a high computational cost that is disproportionate to what he wants to get. This calls for *constraint-based association rule mining*.

BACKGROUND

Intuitively, *constraint-based association rule mining* aims to develop a systematic method by which the user can find important association among items in a database of transactions. By doing so, the user can then figure out how the presence of some *interesting* items (i.e., items that are interesting to the user) implies the presence of other interesting items in a transaction. To elaborate, many retailers, such as supermarkets, carry a large number of items. Progress in bar-code technology has made it possible to record items purchased on a per-transaction basis. Each customer purchases one or more items in a given transaction. Types and quantities of different items can vary significantly among transactions and/or customers. Given a database of sales transactions, constraint-based association rule mining helps discover important relationships between the different interesting items so that retailers can learn how the presence of some interesting items in a transaction relates to the presence of other interesting items in the same transaction. The discovered rules reveal the buying patterns in consumer behaviour. These rules are useful in making decisions in applications such as customer targeting, shelving, and sales promotions. Although we describe this problem in the context of the shoppers' market basket application, constraint-based association rule mining is also useful in many other applications such as finding important relationships from financial time series, Web click streams, and biomedical records. When compared with its traditional unconstrained counterpart, constraint-based association rule mining allows the user to express his interest via the use of constraints. By exploiting some nice properties of these constraints, the user can efficiently find association rules that are interesting to him.

More formally, the problem of *constraint-based association rule mining* can be described as follows. Given a database of transactions, each transaction corresponds to a set of items (also known as an *itemset*) that appear together (say, merchandise items that are purchased together by a customer in a single visit to a checkout counter). Constraint-based association rule mining generally consists of two key steps. First, it finds interesting frequent itemsets (i.e., frequent itemsets that satisfy user-specified constraints) from the database of transactions. An itemset is frequent if its frequency exceeds or equals the user-specified minimum frequency threshold. Then, it uses these interesting frequent itemsets to form association rules that satisfy user-specified constraints. Typically, rules are of the form "A➔C" such that both A (which represents the antecedent of the rule) and C (which represents the consequent of the rule) are interesting frequent itemsets.

MAIN FOCUS

Constraint-based association rule mining generally aims to mine association rules that satisfy user-specified constraints, where the antecedent and the consequent of the rules are frequent itemsets that satisfy user-specified constraints. It has several advantages over its traditional unconstrained counterpart. First, it provides *user flexibility* so that the user is able to express his interest by specifying various types of constraints. Second, it leads to *system optimization* so that the computational cost for rules is proportionate to what the user wants to get. Note that, on the surface, it may appear that constraint checking would incur extra computation. However, the constraints can be pushed deep inside the mining process through the exploitation of their nice properties, and thus reducing computation. In the following, we describe *what* types of constraints have been proposed and we also discuss *how* the properties of constraints can be exploited to efficiently find association rules that satisfy the constraints.

Types of Constraints

The user-specified constraints can be categorized according to their types or according to their properties. For constraint-based association rule mining, the user can specify various types of constraints—which include knowledge-type constraints, data constraints, dimension constraints, level constraints, interestingness constraints, and rule constraints (Han & Kamber, 2006). Let us give a brief overview of these types of user-specified constraints as follows:

- **Knowledge-type constraints** allow the user to specify what type of knowledge (e.g., association, correlation, causality) to be discovered.
- **Data constraints** allow the user to specify what set of data (e.g., sales transactions, financial time series, Web click streams, biomedical records) to be used in the mining process.
- **Dimension constraints** allow the user to specify how many dimensions of data to be used when forming rules. By specifying dimension constraints, the user could express his interest of finding one-dimensional rules (e.g., "buy(milk) ➔ buy(bread)" that involves only one dimension "buy"), two-dimensional rules (e.g., "occupation(student) ➔ buy(textbook)" that relates two dimensions "occupation" and "buy"), or multi-dimensional rules.
- **Level constraints** allow the user to specify how many levels of the concept hierarchy to be used in the mining process. By specifying level constraints, the user could express his interest of finding single-level rules (e.g., "milk➔bread" that involves only a single level of the concept hierarchy) or multi-level rules (e.g., "dairy product ➔ Brand-X white bread" that involves multiple levels as (1) milk is a dairy product and (2) the consequent of this rule is a brand of white bread, which in turn is a kind of bread).
- **Interestingness constraints** allow the user to specify what statistical measure (e.g., support, confidence, lift) or thresholds to be applied when computing the interestingness of rules.
- **Rule constraints** allow the user to specify what forms of rules (e.g., what items to be included in or excluded from the rules) to be mined.

Over the past decade, several specific constraints have been proposed for constraint-based association rule mining. The following are some notable examples of rule constraints. For instance, Srikant et al. (1997) considered *item constraints*, which allow the user to impose a Boolean expression over the presence or absence of items in the association rules. The item constraint "(jackets AND shoes) OR (shirts AND (NOT hiking boots))" expresses the user interest of finding rules that either contain jackets and shoes or contain shirts but not hiking boots.

Lakshmanan, Ng, and their colleagues (Ng et al., 1998; Lakshmanan et al., 1999, 2003) proposed a constraint-based association rule mining framework, within which the user can specify a rich set of rule constraints. These constraints include SQL-style aggregate constraints and non-aggregate constraints like domain constraints. *SQL-style aggregate constraints* are of the form "*agg*(S.*attribute*) θ *constant*", where *agg* is an SQL-style aggregate function (e.g., min, max, sum, avg) and θ is a Boolean comparison operator (e.g., =, ≠, <, ≤, ≥, >). For example, the aggregate constraint "min(S.Price) ≥ 20" expresses that the minimum price of all items in an itemset S is at least $20. *Domain constraints* are non-aggregate constraints, and they can be of the following forms: (1) "S.*attribute* θ *constant*", where θ is a Boolean comparison operator; (2) "*constant* ∈ S.*attribute*"; (3) "*constant* ∉ S.*attribute*";

or (4) "S.*attribute* φ *set of constants*", where φ is a set comparison operator (e.g., $=, \neq, \subset, \not\subset, \subseteq, \supseteq, \supset$). For example, the domain constraint "S.Type = snack" expresses that each item in an itemset S is snack. Other examples include domain constraints "10 $\in$ S.Price" (which expresses that a $10-item must be in an itemset S), "300 $\notin$ S.Price" (which expresses that an itemset S does not contain any $300-item), and "S.Type $\supseteq$ {snack, soda}" (which expresses that an itemset S must contain some snacks and soda).

In addition to allowing the user to impose these aggregate and non-aggregate constraints on the antecedent as well as the consequent of the rules, Lakshmanan et al. (1999) also allowed the user to impose *two-variable constraints* (i.e., constraints that connect two variables representing itemsets). For example, the two-variable constraint "max(A.Price) $\leq$ min(C.Price)" expresses that the maximum price of all items in an itemset A is no more than the minimum price of all items in an itemset C, where A and C are itemsets in the antecedent and the consequent of a rule respectively.

Gade et al. (2004) mined closed itemsets that satisfy the *block constraint*. This constraint determines the significance of an itemset S by considering the dense block formed by items within S and by transactions associating with S.

Properties of Constraints

A simple way to find association rules satisfying the aforementioned user-specified constraints is to perform a post-processing step (i.e., test to see if each association rule satisfies the constraints). However, a better way is to exploit properties of constraints so that the constraints can be pushed inside the mining process, which in turn make the computation for association rules satisfying the constraints be proportionate to what the user wants to get. Hence, besides categorizing the constraints according to their types, constraints can also be categorized according to their properties—which include anti-monotonic constraints, succinct constraints, monotonic constraints, and convertible constraints.

Lakshmanan, Ng, and their colleagues (Ng et al., 1998; Lakshmanan et al., 2003) developed the CAP and the DCF algorithms in the constraint-based association rule mining framework mentioned above. Their algorithms exploit properties of anti-monotonic constraints to give as much pruning as possible. A constraint such as "min(S.Price) $\geq$ 20" (which expresses that the minimum price of all items in an itemset S is at least $20) is *anti-monotonic* because any itemset S violating this constraint implies that all its supersets also violate the constraint. Note that, for any itemsets S & Sup such that S $\subseteq$ Sup, it is always the case that min(Sup.Price) $\leq$ min(S.Price). So, if the minimum price of all items in S is less than $20, then the minimum price of all items in Sup (which is a superset of S) is also less than $20. By exploiting the property of anti-monotonic constraints, both CAP and DCF algorithms can safely remove all the itemsets that violate anti-monotonic constraints and do not need to compute any superset of these itemsets.

Note that the constraint "support(S) $\geq$ minsup" commonly used in association rule mining (regardless whether it is constraint-based or not) is also *anti-monotonic*. The Apriori, CAP and DCF algorithms all exploit the property of such an anti-monotonic constraint in a sense that they do not consider supersets of any infrequent itemset (because these supersets are guaranteed to be infrequent).

In addition to anti-monotonic constraints, both CAP and DCF algorithms (in the Apriori-based framework) also exploit succinct constraints. Moreover, Leung et al. (2002) proposed a constraint-based mining algorithm—called FPS—to find frequent itemsets that satisfy succinct constraints in a tree-based framework. A constraint "10 $\in$ S.Price" (which expresses that a $10-item must be in an itemset S) is *succinct* because one can directly generate precisely *all* and *only* those itemsets satisfying this constraint by using a precise "formula". To elaborate, all itemsets satisfying "10 $\in$ S.Price" can be precisely generated by combining a $10-item with some other (optional) items of any prices. Similarly, the constraint "min(S.Price) $\geq$ 20" is also succinct because all itemsets satisfying this constraint can be precisely generated by combining items of price at least $20. By exploiting the property of succinct constraints, the CAP, DCF and FPS algorithms use the "formula" to directly generate precisely *all* and *only* those itemsets satisfying the succinct constraints. Consequently, there is no need to generate and then exclude itemsets not satisfying the constraints. Note that a succinct constraint can also be anti-monotonic (e.g., "min(S.Price) $\geq$ 20").

Grahne et al. (2000) exploited monotonic constraints when finding correlated itemsets. A constraint such as "10 $\in$ S.Price" is *monotonic* because any itemset S

satisfying this constraint implies that all supersets of S also satisfy the constraint. By exploiting the property of monotonic constraints, one does not need to perform further constraint checking on any superset of an itemset S once S is found to be satisfying the monotonic constraints. Note that this property deals with satisfaction of constraints, whereas the property of anti-monotonic constraints deals with violation of constraints (i.e., any superset of an itemset violating anti-monotonic constraints must also violate the constraints).

Bonchi and Lucchese (2006) also exploited monotonic constraints, but they did so when mining closed itemsets. Bucila et al. (2003) proposed a dual mining algorithm—called DualMiner—that exploits both anti-monotonic and monotonic constraints simultaneously to find itemsets satisfying the constraints.

Knowing that some tough constraints such as "sum(S.Price) $\leq$ 70" (which expresses that the total price of all items in an itemset S is at most \$70) are not anti-monotonic or monotonic in general, Pei et al. (2004) converted these constraints into ones that possess anti-monotonic or monotonic properties. For example, when items within each itemset are arranged in ascending order of prices, for any *ordered* itemsets S & Sup, if S is a prefix of Sup, then Sup is formed by adding to S those items that are more expensive than any item in S (due to the ascending price order). So, if S violates the constraint (i.e., the total price of all items in S exceeds \$70), then so does Sup (i.e., the total price of all items in Sup also exceeds \$70). Such a constraint is called *convertible anti-monotonic constraint*.

Similarly, by arranging items in ascending order of prices, the constraint "sum(S.Price) $\geq$ 60" possesses monotonic properties (i.e., if an itemset S satisfies this constraint, then so does any *ordered* itemset having S as its prefix). Such a constraint is called *convertible monotonic constraint.* To a further extent, Bonchi and Lucchese (2007) exploited other tough constraints involving aggregate functions variance and standard deviation.

FUTURE TRENDS

A future trend is to integrate constraint-based association rule mining with other data mining techniques. For example, Lakshmanan et al. (2003) and Leung (2004b) integrated *interactive mining* with constraint-based association rule mining so that the user can interactively monitor the mining process and change the mining parameters. As another example, Leung (2004a) proposed a *parallel mining* algorithm for finding constraint-based association rules.

Another future trend is to conduct constraint-based mining on different kinds of data. For example, Leung and Khan (2006) applied constraint-based mining on *data streams*. Pei et al. (2007) and Zhu et al. (2007) applied constraint-based mining on *time sequences* and on *graphs*, respectively.

CONCULSION

Constraint-based association rule mining allows the user to get association rules that satisfy user-specified constraints. There are various types of constraints such as rule constraints. To efficiently find association rules that satisfy these constraints, many mining algorithms exploited properties of constraints such as anti-monotonic, succinct, and monotonic properties of constraints. In general, constraint-based association rule mining is not confined to finding interesting association from market basket data. It can also be useful in many other real-life applications. Moreover, it can be applied on various kinds of data and/or integrated with different data mining techniques.

REFERENCES

Agrawal, R., Imielinski, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. In *Proceedings of the SIGMOD 1993*, 207-216.

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules in large databases. In *Proceedings of the VLDB 1994*, 487-499.

Bonchi, F., & Lucchese, C. (2006). On condensed representations of constrained frequent patterns. *Knowledge and Information Systems, 9*(2), 180-201.

Bonchi, F., & Lucchese, C. (2007). Extending the state-of-the-art of constraint-based pattern discovery. *Data & Knowledge Engineering, 60*(2), 377-399.

Bucila, C., Gehrke, J., Kifer, D., & White, W. (2003). DualMiner: a dual-pruning algorithm for itemsets with constraints. *Data Mining and Knowledge Discovery*,

7(3), 241-272.

Gade, K., Wang, J., & Karypis, G. (2004). Efficient closed pattern mining in the presence of tough block constraints. In *Proceedings of the KDD 2004*, 138-147.

Grahne, G., Lakshmanan, L.V.S., & Wang, X. (2000). Efficient mining of constrained correlated sets. In *Proceedings of the ICDE 2000*, 512-521.

Han, J., & Kamber, M. (2006). *Data Mining: Concepts and Techniques, 2nd ed.* San Francisco, CA, USA: Morgan Kaufmann Publishers.

Han, J., Pei, J., & Yin, Y. (2000). Mining frequent patterns without candidate generation. In *Proceedings of the SIGMOD 2000*, 1-12.

Korn, F., Labrinidis, A., Kotidis, Y., & Faloutsos, C. (1998). Ratio rules: a new paradigm for fast, quantifiable data mining. In *Proceedings of the VLDB 1998*, 582-593.

Lakshmanan, L.V.S., Leung, C.K.-S., & Ng, R.T. (2003). Efficient dynamic mining of constrained frequent sets. *ACM Transactions on Database Systems*, *28*(4), 337-389.

Lakshmanan, L.V.S., Ng, R., Han, J., & Pang, A. (1999). Optimization of constrained frequent set queries with 2-variable constraints. In *Proceedings of the SIGMOD 1999*, 157-168.

Leung, C.K.-S. (2004a). Efficient parallel mining of constrained frequent patterns. In *Proceedings of the HPCS 2004*, 73-82.

Leung, C.K.-S. (2004b). Interactive constrained frequent-pattern mining system. In *Proceedings of the IDEAS 2004*, 49-58.

Leung, C.K.-S., & Khan, Q.I. (2006). Efficient mining of constrained frequent patterns from streams. In *Proceedings of the IDEAS 2006*, 61-68.

Leung, C.K.-S., Lakshmanan, L.V.S., & Ng, R.T. (2002). Exploiting succinct constraints using FP-trees. *SIGKDD Explorations*, *4*(1), 40-49.

Ng, R.T., Lakshmanan, L.V.S., Han, J., & Pang, A. (1998). Exploratory mining and pruning optimizations of constrained associations rules. In *Proceedings of the SIGMOD 1998*, 13-24.

Park, J.S., Chen, M.-S., & Yu, P.S. (1997). Using a hash-based method with transaction trimming for mining association rules. *IEEE Transactions on Knowledge and Data Engineering*, *9*(5), 813-825.

Pei, J., Han, J., & Lakshmanan, L.V.S. (2004). Pushing convertible constraints in frequent itemset mining. *Data Mining and Knowledge Discovery*, *8*(3), 227-252.

Pei, J., Han, J., & Wang, W. (2007). Constraint-based sequential pattern mining: the pattern-growth methods. *Journal of Intelligent Information Systems*, *28*(2), 133-160.

Silverstein, C., Brin, S., & Motwani, R. (1998). Beyond market baskets: generalizing association rules to dependence rules. *Data Mining and Knowledge Discovery*, *2*(1), 39-68.

Srikant, R., Vu, Q., & Agrawal, R. (1997). Mining association rules with item constraints. In *Proceedings of the KDD 1997*, 67-73.

Zhu, F., Yan, X., Han, J., & Yu, P.S. (2007). gPrune: a constraint pushing framework for graph pattern mining. In *Proceedings of the PAKDD 2007*, 388-400.

KEY TERMS

Anti-Monotonic Constraint: A constraint C is *anti-monotonic* if and only if an itemset S satisfying C implies that any subset of S also satisfies C.

Closed Itemset: An itemset S is *closed* if there does not exist any proper superset of S that shares the same frequency with S.

Convertible Anti-Monotonic Constraint: A constraint C is *convertible anti-monotonic* provided there is an order R on items such that an ordered itemset S satisfying C implies that any prefix of S also satisfies C.

Convertible Constraint: A constraint C is *convertible* if and only if C is a convertible anti-monotonic constraint or a convertible monotonic constraint.

Convertible Monotonic Constraint: A constraint C is *convertible monotonic* provided there is an order R on items such that an ordered itemset S violating C implies that any prefix of S also violates C.

Correlated Itemset: An itemset S consisting of k items is *correlated* if any of the 2^k combinations of the items in S and their complements is not independent.

Frequent Itemset: An itemset is *frequent* if its frequency exceeds or equals the user-specified minimum threshold.

Itemset: An *itemset* is a set of items.

Monotonic Constraint: A constraint C is *monotonic* if and only if an itemset S violating C implies that any subset of S also violates C.

Succinct Constraint: A constraint C is *succinct* provided that all itemsets satisfying C can be expressed in terms of powersets of a fixed number of succinct sets using the set union and/or set difference operators. A succinct set is an itemset, in which items are selected from the domain using the usual SQL selection operator. In simple terms, a constraint C is *succinct* meaning that all and only those itemsets satisfying C can be explicitly and precisely generated using some precise "formula".

Constraint-Based Pattern Discovery

Francesco Bonchi
ISTI-C.N.R., Itay

INTRODUCTION

Devising fast and scalable algorithms, able to crunch huge amount of data, was for many years one of the main goals of data mining research. But then we realized that this was not enough. It does not matter how efficient such algorithms can be, the results we obtain are often of limited use in practice. Typically, the knowledge we seek is in a small pool of local patterns hidden within an ocean of irrelevant patterns generated from a sea of data. Therefore, it is the volume of the results itself that creates a second order mining problem for the human expert. This is, typically, the case of association rules and frequent itemset mining (Agrawal & Srikant, 1994), to which, during the last decade a lot of researchers have dedicated their (mainly algorithmic) investigations. The computational problem is that of efficiently mining from a database of transactions, those itemsets which satisfy a user-defined constraint of minimum frequency.

Recently the research community has turned its attention to more complex kinds of frequent patterns extracted from more structured data: *sequences*, *trees*, and *graphs*. All these different kinds of pattern have different peculiarities and application fields, but they all share the same computational aspects: a usually very large input, an exponential search space, and a too large solution set. This situation—too many data yielding too many patterns—is harmful for two reasons. First, performance degrades: mining generally becomes inefficient or, often, simply unfeasible. Second, the identification of the fragments of interesting knowledge, blurred within a huge quantity of mostly useless patterns, is difficult. The paradigm of *constraint-based pattern mining* was introduced as a solution to both these problems. In such paradigm, it is the user who specifies to the system what is *interesting* for the current application: constraints are a tool to drive the mining process towards potentially interesting patterns, moreover they can be pushed deep inside the mining algorithm in order to fight the exponential search space curse, and to achieve better performance (Srikant et al., 1997; Ng et al. 1998; Han et al., 1999; Grahne et al., 2000).

BACKGROUND

Intuitively the constraint-based pattern mining problem requires to extract from a database the patterns which satisfy a conjunction of constraints. Such conjunction usually contains the minimum frequency constraint, plus other constraints which can be defined on the structure of the patterns (e.g., on the size of the patterns, or on the singletons that the patterns may or may not contain), or on some aggregated properties of the patterns (e.g., the sum of "prices", or the average of "weights").

The following is an example of constraint-based mining query:

$$Q : supp_D(X) \geq 1500 \wedge |X| \geq 5 \wedge avg(X.weight) \leq 15 \wedge sum(X.price) \geq 22$$

it requires to mine, from database D, all patterns which are frequent (have a support at least 1500), have cardinality at least 5, have average weight at most 15 and a sum of prices at least 22.

The constraint-based mining paradigm has been successfully applied in medical domain (Ordonez et al., 2001), and in biological domain (Besson et al., 2005). According to the constraint-based mining paradigm, the data analyst must have a high-level vision of the pattern discovery system, without worrying about the details of the computational engine, in the very same way a database designer has not to worry about query optimization: she must be provided with a set of primitives to declaratively specify to the pattern discovery system how the interesting patterns should look like, i.e., which conditions they should obey. Indeed, the task of composing all constraints and producing the most efficient mining strategy (execution plan) for the given data mining query, should be left to an underlying *query optimizer*. Therefore, constraint-based frequent pattern mining has been seen as a query optimization problem

(Ng et al., 1998), i.e., developing efficient, sound and complete evaluation strategies for constraint-based mining queries.

Among all the constraints, the frequency constraint is computationally the most expensive to check, and many algorithms, starting from Apriori, have been developed in the years for computing patterns which satisfy a given threshold of minimum frequency (see the FIMI repository http://fimi.cs.helsinki.fi for the state-of-the-art of algorithms and softwares). Therefore, a naïve solution to the constraint-based frequent pattern problem could be to first mine all frequent patterns and then test them for the other constraints satisfaction. However more efficient solutions can be found by analyzing the properties of constraints comprehensively, and exploiting such properties in order to push constraints in the frequent pattern computation. Following this methodology, some classes of constraints which exhibit nice properties (e.g., monotonicity, anti-monotonicity, succinctness, etc) have been individuated in the last years, and on the basis of these properties efficient algorithms have been developed.

MAIN RESULTS

A first work defining classes of constraints which exhibit nice properties is Ng et al. (1998). In that work is introduced an Apriori-like algorithm, named CAP, which exploits two properties of constraints, namely *anti-monotonicity* and *succinctness*, in order to reduce the frequent itemsets computation. Four classes of constraints, each one with its own associated computational strategy, are identified:

1. constraints that are anti-monotone but not succinct;
2. constraints that are both anti-monotone and succinct;
3. constraints that are succinct but not anti-monotone;
4. constraints that are neither.

Anti-Monotone and Succinct Constraints

An anti-monotone constraint is such that, if satisfied by a pattern, it is also satisfied by all its subpatterns. The frequency constraint is the most known example of a anti-monotone constraint. This property, *the anti-monotonicity of frequency*, is used by the Apriori (Agrawal & Srikant, 1994) algorithm with the following heuristic: if a pattern X does not satisfy the frequency constraint , then no super-pattern of X can satisfy the frequency constraint, and hence they can be pruned. This pruning can affect a large part of the search space, since itemsets form a lattice. Therefore the Apriori algorithm operates in a level-wise fashion moving bottom-up, level-wise, on the patterns lattice, from small to large itemsets, generating the set of *candidate patterns* at iteration k from the set of frequent patterns at the previous iteration. This way, each time it finds an infrequent pattern it implicitly prunes away all its supersets, since they will not be generated as candidate itemsets. Other anti-monotone constraints can easily be pushed deeply down into the frequent patterns mining computation since they behave exactly as the frequency constraint: if they are not satisfiable at an early level (small patterns), they have no hope of becoming satisfiable later (larger patterns). Conjoining other anti-monotone constraints to the frequency one we just obtain a more selective anti-monotone constraint. As an example, if "price" has positive values, then the constraint $sum(X.price) \leq 50$ is anti-monotone. Trivially, let the pattern X be *{olive_oil, tomato_can, pasta, red_wine}* and suppose that it satisfies such constraints, then any of its sub-patterns will satisfy the constraint as well: for instance the set *{olive_oil, red_wine}* for sure will have a sum of prices less than 50. On the other hand, if X does not satisfy the constraint, then it can be pruned since none of its supersets will satisfy the constraint.

Informally, a succinct constraint is such that, whether a pattern X satisfies it or not, can be determined based on the basic elements of X. Succinct constraints are said to be *pre-counting pushable*, i.e., they can be satisfied at candidate-generation time just taking into account the pattern and the single items satisfying the constraint. These constraints are pushed in the level-wise computation by adapting the usual *candidate-generation* procedure of the Apriori algorithm, w.r.t. the given succinct constrain, in such a way that it prunes every pattern which does not satisfy the constraint and that it is not a sub-pattern of any further valid pattern.

Constraints that are both anti-monotone and succinct can be pushed completely in the level-wise computation before it starts (at pre-processing time). For instance, consider the constraint $min(X.price) \geq v$. It is straightforward to see that it is both anti-monotone and succinct. Thus, if we start with the first set of candidates

formed by all singleton items having price greater than *v*, during the computation we will generate all and only those patterns satisfying the given constraint. Constraints that are neither succinct nor anti-monotone are pushed in the CAP computation by inducing weaker constraints which are either anti-monotone and/or succinct. Consider the constraint $avg(X.price) \leq v$ which is neither succinct nor anti-monotone. We can push the weaker constraint $min(X.price) \leq v$ with the advantage of reducing the search space and the guarantee that at least all the valid patterns will be generated.

Monotone Constraints

Monotone constraints work the opposite way of anti-monotone constraints. Whenever a pattern satisfies a monotone constraint, so will do all its super-patterns (or the other way around: if a pattern does not satisfy a monotone constraint, none of its sub-pattern will satisfy the constraint as well). The constraint $sum(X.price) \geq 500$ is monotone, since all prices are not negative. Trivially, if a pattern X satisfies such constraint, then any of its super-patterns will satisfy the constraint as well. Since the frequency computation moves from small to large patterns, we can not push such constraints in it straightforwardly. At an early stage, if a pattern is too small or too cheap to satisfy a monotone constraint, we can not yet say nothing about its supersets. Perhaps, just adding a very expensive single item to the pattern could raise the total sum of prices over the given threshold, thus making the resulting pattern satisfy the monotone constraint. Many studies (De Raedt & Kramer, 2001; Bucila et al., 2002; Boulicaut & Jeudy, 2002; Bonchi et al., 2003a) have attacked this computational problem focussing on its search space, and trying some smart exploration of it. For example, Bucila et al. (2002) try to explore the search space from the top and from the bottom of the lattice in order to exploit at the same time the symmetric behavior of monotone and anti-monotone constraints. Anyway, all of these approaches face the inherent difficulty of the computational problem: the *tradeoff* existing between anti-monotone and monotone constraints, that can be described as follows. Suppose that a pattern has been removed from the search space because it does not satisfy a monotone constraint. This pruning avoids the expensive frequency check for this pattern, but on the other hand, if we check its frequency and find it smaller than the frequency threshold, we may prune away all its super-patterns, thus saving the frequency check for all of them. In other words, by monotone pruning we risk to lose anti-monotone pruning opportunities given by the pruned patterns. The tradeoff is clear: pushing monotone constraints can save frequency tests, however the results of these tests could have lead to more effective anti-monotone pruning. In Bonchi et al. (2003b) a completely new approach to exploit monotone constraints by means of data-reduction is introduced. The *ExAnte Property* is obtained by shifting attention from the pattern search space to the input data. Indeed, the *tradeoff* exists only if we focus exclusively on the search space of the problem, while if exploited properly, monotone constraints can reduce dramatically the data in input, in turn strengthening the anti-monotonicity pruning power. The ExAnte property states that a transaction which does not satisfy the given monotone constraint can be deleted from the input database since it will never contribute to the support of any pattern satisfying the constraint. A major consequence of removing transactions from input database in this way, is that it implicitly reduces the support of a large amount of patterns that do not satisfy the monotone constraint as well, resulting in a reduced number of candidate patterns generated during the mining algorithm. Even a small reduction in the database can cause a huge cut in the search space, because all super-patterns of infrequent patterns are pruned from the search space as well. In other words, monotonicity-based data-reduction of transactions strengthens the anti-monotonicity-based pruning of the search space. This is not the whole story, in fact, singleton items may happen to be infrequent after the pruning and they can not only be removed from the search space together with all their supersets, but for the same anti-monotonicity property they can be deleted also from all transactions in the input database. Removing items from transactions brings another positive effect: reducing the size of a transaction which satisfies the monotone constraint can make the transaction violate it. Therefore a growing number of transactions which do not satisfy the monotone constraint can be found and deleted. Obviously, we are inside a loop where two different kinds of pruning cooperate to reduce the search space and the input dataset, strengthening each other step by step until no more pruning is possible (a fix-point has been reached). This is the key idea of the ExAnte pre-processing method. In the end, the reduced dataset resulting from this fix-point computation is usually much smaller than the initial dataset, and it can

feed any frequent itemset mining algorithm for a much smaller (but complete) computation. This simple yet very effective idea has been generalized from pre-processing to effective mining in two main directions: in an Apriori-like breadth-first computation in *ExAMiner* (Bonchi et al., 2003c), and in a depth-first computation in *FP-Bonsai* (Bonchi & Goethals, 2004).

Convertible Constraints

In Pei and Han (2000); Pei et al. (2001) the class of convertible constraints is introduced, and depth-first strategy, based on a prefix-tree data structure, able to push such constraints is proposed. Consider the constraint defined on the average aggregate (e.g., *avg(X: price)* $\geq$ *15*): it is quite straightforward to show that it is not anti-monotone, nor monotone, nor succinct. Subsets (or supersets) of a valid pattern could well be invalid and vice versa. For this reason, the authors state that within the level-wise Apriori framework, no direct pruning based on such constraints can be made. But, if we change the focus from the subset to the prefix concept we can find interesting properties. Consider the constraint *avg(X:price)* $\geq$ *15*, if we arrange the items in price-descending-order we can observe an interesting property: the average of a pattern is no more than the average of its prefix, according to this order.

As an example consider the following items sorted in price-descending-order (we use the notation *[item : price]*): *[a : 20], [b : 15], [c : 8], [d : 8], [e : 4]*. We got that the pattern *{a,b,c}* violates the constraint, thus we can conclude that any other pattern having *{a,b,c}* as prefix will violate the constraint as well: the constraint has been converted to an anti-monotone one!

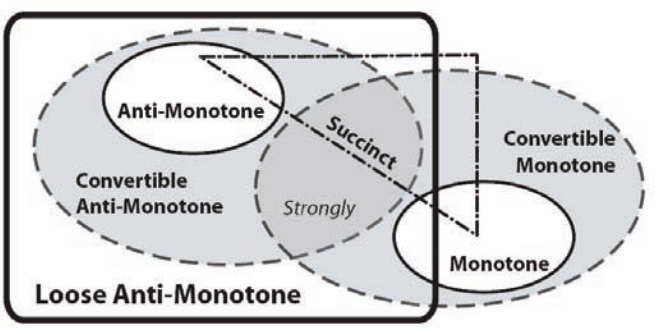
Therefore if we adopt a depth-first visit of the search space, based on the prefix, we can exploit this kind of constraints. More formally, a constraint is said to be convertible anti-monotone provided there is an order on items such that whenever an itemset X satisfies the constraint , so does any prefix of X. A constraint is said to be convertible monotone provided there is an order R on items such that whenever an itemset X violates the constraint, so does any prefix of X.

A major limitation of this approach is that the initial database (internally compressed in the prefix-tree) and all intermediate projected databases must fit into main memory. If this requirement cannot be met, this approach can simply not be applied anymore. This problem is very hard when using an order on items different from the frequency-based one, since it makes the prefix-tree lose its compressing power. Thus we have to manage much greater data structures, requiring a lot more main memory which might not be available. Another important drawback of this approach is that it is not possible to take full advantage of a conjunction of different constraints, since each constraint could require a different ordering of items.

Non-Convertible Constraints

A first work, trying to address the problem of how to push constraints which are not convertible, is Kifer et al. (2003). The framework proposed in that paper is based on the concept of finding a *witness*, i.e., a pattern such that, by testing whether it satisfies the constraint we can deduce information about properties of other patterns, that can be exploited to prune the search space. This idea is embedded in a depth-first visit of the patterns

Figure 1.



search space. The authors instantiate their framework to the constraint based on the *variance* aggregate. Another work going beyond convertibility is Bonchi & Lucchese (2005), where a new class of constraints sharing a nice property named *"loose anti-monotonicity"*, is individuated. This class is a proper superclass of convertible anti-monotone constraints, and it can also deal with other tougher constraints. Based on loose anti-monotonicity the authors define a data reduction strategy, which makes the mining task feasible and efficient. Recall that an anti-monotone constraint is such that, if satisfied by a pattern then it is satisfied by *all* its sub-patterns. Intuitively a loose anti-monotone constraint is such that, if it is satisfied by an itemset of cardinality k then it is satisfied by *at least one* of its subsets of cardinality k-1. It is quite straightforward to show that the constraints defined over variance, standard deviation or similar aggregated measures, are not convertible but are loose anti-monotone. This property can be exploited within a levelwise Apriori framework in the following way: at the k_{th} iteration a transaction is not superset of at least one pattern of size k which satisfies both the frequency and the loose anti-monotone constraint, then the transaction can be deleted from the database. As in ExAMiner (Bonchi et al., 2003c) the anti-monotonicity based data reduction was coupled with the monotonicity based data reduction, similarly we can embedd the loose anti-monotonicity based data reduction described above within the general Apriori framework. The more data-reduction techniques the better: we can exploit them all together, as they strengthen each other and strengthen search space pruning as well; i.e. and the total benefit is always greater than the sum of the individual benefits.

FUTURE TRENDS

So far the research on constraint-based pattern discovery has mainly focused on developing efficient mining techniques. There are still many aspects which deserve further investigations, especially going towards practical systems and methodologies of use for the paradigm of pattern discovery based on constraints. One important aspect is the *incremental mining* of constraint-based queries; i.e., the reusing of already mined solution sets for new queries computation. Altough few studies (Cong & Liu, 2002; Cong et al., 2004) have started investigating this direction there is still room for improvement. Another important aspect is to study and define the relationships between constraint-based mining queries and *condensed representations*: once again, little work already exists (Boulicaut & Jeudy, 2002; Bonchi & Lucchese, 2004), but the topic is worth of further investigation. Finally we must design and develop *practical systems* able to use the constraint-based techniques on real-world problems.

CONCLUSION

In this chapter we have introduced the paradigm of pattern discovery based on user-defined constraints. Constraints are a tool in the user's hands to drive the discovery process toward interesting knowledge, while at the same time reducing the search space and thus the computation time. We have reviewed the state-of-the-art of constraints that can be pushed in the frequent pattern mining process and their associated computational startegies. Finally we have indicated few path of future research and development.

REFERENCES

Agrawal R. & Srikant R. (1994) Fast Algorithms for Mining Association Rules in Large Databases. *In Proceedings of the 20th International Conference on Very Large Databases,* VLDB 1994, Santiago de Chile, Chile.

Besson J., Robardet C., Boulicaut J.F. & Rome S. (2005) Constraint-based concept mining and its application to microarray data analysis. *Intelligent Data Analysis Journal*, Vol. 9(1): pp 59–82.

Bonchi F. & Goethals B. (2004) FP-Bonsai: The Art of Growing and Pruning Small FP-Trees. In *Proceedings of Advances in Knowledge Discovery and Data Mining, 8th Pacific-Asia Conference*, PAKDD 2004, Sydney, Australia.

Bonchi F & Lucchese C. (2004) On Closed Constrained Frequent Pattern Mining. In *Proceedings of the Third IEEE International Conference on Data Mining*, ICDM 2004, Brighton, UK.

Bonchi F. & Lucchese C. (2005). Pushing Tougher Constraints in Frequent Pattern Mining. In *Proceedings of Advances in Knowledge Discovery and Data Mining, 9th Pacific-Asia Conference*, PAKDD 2005, Hanoi, Vietnam.

Bonchi F., Giannotti F., Mazzanti A. & Pedreschi D. (2003a). Adaptive Constraint Pushing in Frequent Pattern Mining. In *Proceedings of the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases*, PKDD 2003, Cavtat-Dubrovnik, Croatia.

Bonchi F., Giannotti F., Mazzanti A. & Pedreschi D. (2003b). ExAnte: Anticipated Data Reduction in Constrained Pattern Mining. In *Proceedings of the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases*, PKDD 2003, Cavtat-Dubrovnik, Croatia.

Bonchi F., Giannotti F., Mazzanti A. & Pedreschi D. (2003c). ExAMiner: optimized level-wise frequent pattern mining with monotone constraints. In *Proceedings of the Third IEEE International Conference on Data Mining*, ICDM 2003, Melbourne, Florida, USA.

Boulicaut J.F. & Jeudy B. (2002). Optimization of Association Rule Mining Queries. *Intelligent Data Analysis Journal* 6(4):341–357.

Bucila C., Gehrke J., Kifer D. & White W. (2002). DualMiner: A Dual-Pruning Algorithm for Itemsets with Constraints. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002, Edmonton, Alberta, Canada.

Cong G. & Liu B. (2002). Speed-up iterative frequent itemset mining with constraint changes. In *Proceedings of the Third IEEE International Conference on Data Mining*, ICDM 2002, Maebashi City, Japan.

Cong G., Ooi B.C., Tan K.L. & Tung A.K.H. (2004). Go green: recycle and reuse frequent patterns. In *Proceedings of the 20th IEEE International Conference on Data Engineering*, ICDE 2004, Boston, USA.

De Raedt L. & Kramer S. (2001). The Levelwise Version Space Algorithm and its Application to Molecular Fragment Finding. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, IJCAI 2001, Seattle, Washington, USA.

Grahne G., Lakshmanan L. & Wang X. (2000). Efficient Mining of Constrained Correlated Sets. In *Proceedings of the 16th IEEE International Conference on Data Engineering*, ICDE 2000, San Diego, California, USA.

Han J., Lakshmanan L. & Ng R. (1999). Constraint-Based, Multidimensional Data Mining. *Computer*, 32(8): pp 46–50.

Kifer D., Gehrke J., Bucila C. & White W. (2003). How to Quickly Find a Witness. In *Proceedings of 2003 ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, PODS 2003, San Diego, CA, USA.

Lakshmanan L., Ng R., Han J. & Pang A. (1999). Optimization of Constrained Frequent Set Queries with 2-variable Constraints. In *Proceedings of 1999 ACM International Conference on Management of Data*, SIGMOD 1999, Philadelphia, Pennsylvania, USA, pp 157–168

Ng R., Lakshmanan L., Han J. & Pang A. (1998). Exploratory Mining and Pruning Optimizations of Constrained Associations Rules. In *Proceedings of 1998 ACM International Conference on Management of Data*, SIGMOD 1998, Seattle, Washington, USA.

Ordonez C et al. (2001). Mining Constrained Association Rules to Predict Heart Disease. In *Proceedings of the First IEEE International Conference on Data Mining*, December 2001, San Jose, California, USA, pp 433–440.

Pei J. & Han J. (2000). Can we push more constraints into frequent pattern mining? In *Proceedings ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD 2000, Boston, MA, USA.

Pei J., Han J. & Lakshmanan L. (2001). Mining Frequent Item Sets with Convertible Constraints. In *Proceedings of the 17th IEEE International Conference on Data Engineering*, ICDE 2001, Heidelberg, Germany.

Srikant R., Vu Q. & Agrawal R. (1997). Mining Association Rules with Item Constraints. In *Proceedings ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD 1997, Newport Beach, California, USA.

KEY TERMS

Anti-Monotone Constraint: A constraint such that, if satisfied by a pattern, it is also satisfied by all its subpatterns.

Constraint-Based Pattern Discovery: When the Pattern Discovery task is driven by a conjunction of user-defined constraints, which provides a description of the interesting patterns the user is looking for.

Constraint-Pushing Techniques: Algorithmic techniques developed to "push" some classes of constraints within the Frequent Pattern Discovery process. These techniques usually exploit constraints to prune the search space or to reduce the data of the frequent pattern computation, achieving better performances.

Convertible Constraint: A constraint that is neither monotone, nor anti-monotone, nor succinct; but that can be *converted* to a monotone or to an anti-monotone one, by imposing an order on the basic elements of patterns.

Frequent Pattern Discovery: The discovery of structures of information frequently (i.e., a number of times larger than a given threshold) occurring within a database. For instance, given a database of sets, the discovery of frequent subsets; given a database of graphs, the discovery of frequent subgraphs.

Loose Anti-Monotone Constraint: A constraint such that, if it is satisfied by a pattern of size k then it is satisfied by *at least one* of its subpatterns of size k-1.

Monotone Constraint: A constraint such that, if satisfied by a pattern, it is also satisfied by all its super-patterns.

Succinct Constraint: A constraint such that, whether a pattern X satisfies it or not, can be determined based on the basic components of X.

Context-Driven Decision Mining

Alexander Smirnov
Institution of the Russian Academy of Sciences, St. Petersburg Institute for Informatics and Automation RAS, Russia

Michael Pashkin
Institution of the Russian Academy of Sciences, St. Petersburg Institute for Informatics and Automation RAS, Russia

Tatiana Levashova
Institution of the Russian Academy of Sciences, St. Petersburg Institute for Informatics and Automation RAS, Russia

Alexey Kashevnik
Institution of the Russian Academy of Sciences, St. Petersburg Institute for Informatics and Automation RAS, Russia

Nikolay Shilov
Institution of the Russian Academy of Sciences, St. Petersburg Institute for Informatics and Automation RAS, Russia

INTRODUCTION

Decisions in the modern world are often made in rapidly changing, sometimes unexpected, situations. Such situations require availability of systems / tools allowing fast and clear description of situation, generation of new and reuse of previously made effective solutions for situation reformation, selection of a right decision maker and supplying him/her with necessary data. Such tools include components for actual situation description, user modeling, finding appropriate methods for problem solving, integration of data from heterogeneous sources, finding / generation of insufficient data, removing uncertainties, estimating solutions, etc. During decision making process a large amount of auxiliary raw data are accumulated in repositories. Methods of data mining are used in such systems for different purposes: finding associative rules between decisions and factors affecting them, user clustering using decision trees and neural networks, recognition of common users' features / interests and others (Chiang et al., 2006; Li, 2005; Thomassey and Fiordaliso, 2006). Validation of the obtained results can be performed using simulation software modules.

BACKGROUND

The chapter presents a developed approach that assumes usage of (i) ontologies for application domain description (notions, relations between notions, data sources and methods), (ii) user profiles to accumulate raw data and build the system's vision of the user, and (iii) context for actual situation description. The developed approach is oriented to producing an ontology-driven context model so that the decision makers would be provided with the information and knowledge required in the situation they are interested in and according to the roles they play.

Within the presented approach the context is a weakly-structured information containing three constituents: (i) ontology elements describing the actual situation; (ii) user data representing user's role, preferences, competences, etc.; and (iii) data / information / knowledge extracted from available sources and relevant the actual situation. Context is built when a situation occurs and used for processing requests related to this situation. The context is a basis for generation and estimation of alternative solutions and presenting them to the decision maker for selecting the best one from his/her point of view. Each situation is described by the following components: context, solutions generated using the context and the final decision (selected solution / solutions). These components are stored in the user profile.

Finding influence of the context on the final decision made by the decision maker playing a certain role is an important task because it helps to (i) find and model typical scenarios of interaction between users; (ii) reveal typical situations within large amount of raw data; and (iii) cluster existing decision makers into groups, thus, allowing reducing the number of supported user

models and increasing the data presentation quality. Finally, these results lead to increasing the decision quality, what is important when decisions have to be made under time pressure. To find the above inference the described here approach applies decision mining techniques as a subarea of data mining.

Analysis of different decisions' kinds is one of the areas of data mining for business processes. The goal of decision mining is to find "rules" explaining under what circumstances certain activity is to be selected rather than the other one. For instance, decision mining aims at detecting data dependencies that affect the routing of a case in the event log of the business process executions (Rozinat and van der Aalst, 2006). There is a set of tools implementing different tasks of decision mining: "Decision Miner" (Rozinat and van der Aalst, 2006), Decision mining software "Risky Business" and "GoldPan" (Decision Mining software, 2007) and other.

Decision mining covers a wide range of problems. Estimation of data quality and interpretation of their semantics is one of the major tasks of decision mining. It requires the following interpretation of data: whether it is relevant, what it actually means, in what units it is measured, etc. Classification of decisions is also one of important tasks for decision mining. These tasks solving requires development of decision models and (semi)automatic decision analysis techniques. For instance, in the approach the concept of decision trees has been adapted to carry out a decision point analysis (Rozinat and van der Aalst, 2006), spatial analysis has been adapted for criminal event prediction (Brown et al., 2006).

The developed approach proposes an application of decision mining techniques to the area of intelligent decision support. Classification of decisions allows discovering correspondence between decision makers and their roles. Revealing preferences of decision makers helps to build decision trees that allow making right decisions in critical situations (semi)automatically.

Main Focus: Context-Sensitive Decision Support

The idea of using ontology-driven context model for decision support arose from the definition. Context is defined as any information that can be used to characterize the situation of an entity where an entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves (Dey et al., 2001).

In modern information systems ontologies are widely used as a basis for domain knowledge description. In the presented approach the formalism of Object-Oriented Constraint Networks (OOCN) is used for a formal ontology representation. Summary of the possibility to convert knowledge elements from OWL (W3C, 2007) as one of the most widespread standards of knowledge representation into the OOCN formalism is presented in Table 1.

The developed approach proposes integration of environmental information and knowledge in context (Smirnov et al., 2005). The context is purposed to represent only relevant information and knowledge from the large amount of those. Relevance of the information and knowledge is evaluated based on what is their relation to an ad hoc problem modeling. The methodology proposes integration of environmental information and domain knowledge in a context of the current situation. It is done through linkage of representation of this knowledge with semantic models of information sources providing information about the environment. The methodology (Figure 1) considers context as a problem model built using knowledge extracted from the application domain and formalized within an ontology by a set of constraints. The set of constraints, additionally to the constraints describing domain knowledge, includes information about the environment and various preferences of the user concerning the problem solving (user defined constraints). Two types of context are used: 1) *abstract context* that is an ontology-based model integrating information and knowledge relevant to the problem, and 2) *operational context* that is an instantiation of the abstract context with data provided by the information sources or calculated based on functions specified in the abstract context.

The approach relies on a three-phase model of decision making process (Simon, 1965). The model describes decision making consisting of "intelligence", "design", and "choice" phases. The proposed approach expands the "intelligence phase" that addresses problem recognition and goal settings into steps reiterating all three phases of this model. Using the constraint satisfaction technology the proposed approach covers the last two phases focusing on a generation of alternative solutions and choice of a decision (Table 2).

The conceptual framework of the developed approach is as follows.

Table 1. Correspondence between OWL and object-oriented constraint networks notations

Element Groups	OWL Elements	
Elements supported by the notation of object-oriented constraint networks	Class complementOf DeprecatedClass DeprecatedProperty disjointWith equivalentClass equivalentProperty maxCardinality	minCardinality Nothing Ontology priorVersion Restriction Thing versionInfo
Elements weakly supported by the notation of object-oriented constraint networks	allValuesFrom AnnotationProperty cardinality DataRange DatatypeProperty	FunctionalProperty incompatibleWith InverseFunctionalProperty OntologyProperty unionOf
Elements not currently supported by the notation of object-oriented constraint networks	hasValue Imports inverseOf	ObjectProperty onProperty
Elements not currently supported by the notation of object-oriented constraint networks, and such support requires additional research	AllDifferent backwardCompatibleWith differentFrom distinctMembers intersectionOf	one of sameAs someValuesFrom SymmetricProperty TransitiveProperty

Figure 1. Context-based decision support

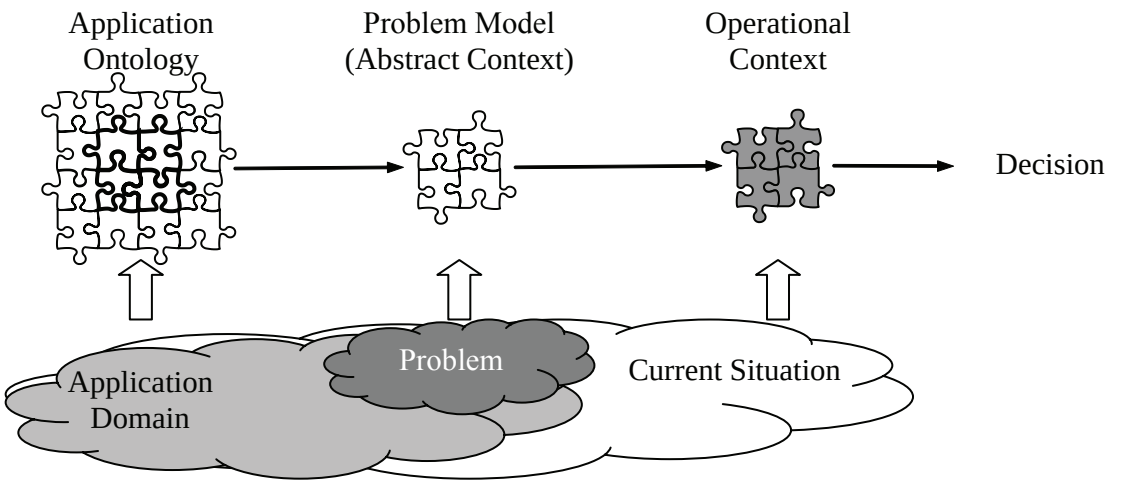


Table 2. Decision making stage related to the three-phase model

Phase	Phase content	Steps	Proposed approach steps
Intelligence	Problem recognition	- fixing goals - setting goals	- abstract context composition - operational context producing
Design	Alternatives generation	- designing alternatives	- constraint-based generating efficient alternatives
Choice	Efficient alternatives selection	- evaluation & choosing alternatives	

Before a problem can be solved an ontology describing relevant areas has to be built. The ontology combines the domain knowledge and problem solving knowledge. Decision making in dynamic domains is characterized by a necessity to dynamically process and integrate information from heterogeneous sources and to provide the user with context-driven assistance for the analysis of the current situation. Systems of context-driven decision making support are based on usage of information / data, documents, knowledge and models for problem identification and solving.

Another necessary prerequisite for the system operation is its connection to information sources. The connection is based on three operations: finding information sources, connection of the found sources to the system using Web-services, and assigning information source references to attributes of the ontology, in other words, defining what sources the ontology attributes take values from.

Once the above operations are completed the decision support system is considered to be ready for problem processing. Personalization is one of important features of decision support. For this reason users are described in the system via user profiles and associated with different roles that set certain limitations to provide only information that is useful for a particular user.

When the user passes the registration in the system the user profile is updated or created if it did not exist before. Then, depending on the user role and available information the user either describes the problem or selects a situation. In the latter case the abstract context is assumed to exist and it is activated. In order to ensure usability of the previously created abstract context the context versioning technique described in the following section is used. In the former case few more steps are required. The problem description stored in the profile for its further analysis is recognized by the system, and recognized meaningful terms are mapped to the vocabulary of the ontology; the vocabulary includes class names, attribute names, string domains, etc. Based on the identified ontology elements and the developed algorithm a set of ontology slices relevant to the problem is built. Based on these slices a set of abstract contexts is generated. Since some of the problem solving methods can be alternative, integration of such slices leads to a set of alternative abstract contexts. The generated abstract contexts are checked for consistency, their attributes are assigned information sources based on the information from the ontology, and it is saved in the context archive. Role constraints are applied to the generated abstract context to remove information that is not interesting for or not supposed to be seen by the user.

The abstract contexts are checked if they are sufficient enough for solving the current problem. Then the required information is acquired from the appropriate information sources and calculations are performed. Due to the chosen OOCN notation a compatible constraint solver can be used for this purpose. The constraint solver analyses information acquired from the sources and produces a set of feasible solutions eliminating contradictory information.

The above operations result in forming operational contexts for each abstract context. Among the operational contexts one with values for all attributes is selected. This operational context is used at the later problem solving stages. All the operational contexts are stored in the context archive, and references to the operational contexts are stored in the user profile.

Based on the identified operational context the set of solutions is generated. Among the generated solutions the user selects the most appropriate one and makes a decision. The solutions and the final decision are stored in the user profile for further analysis. Stored in the profile information can be used for various purposes including audit of user activities, estimation of user skills in certain areas, etc. In the presented research this information is used for decision mining.

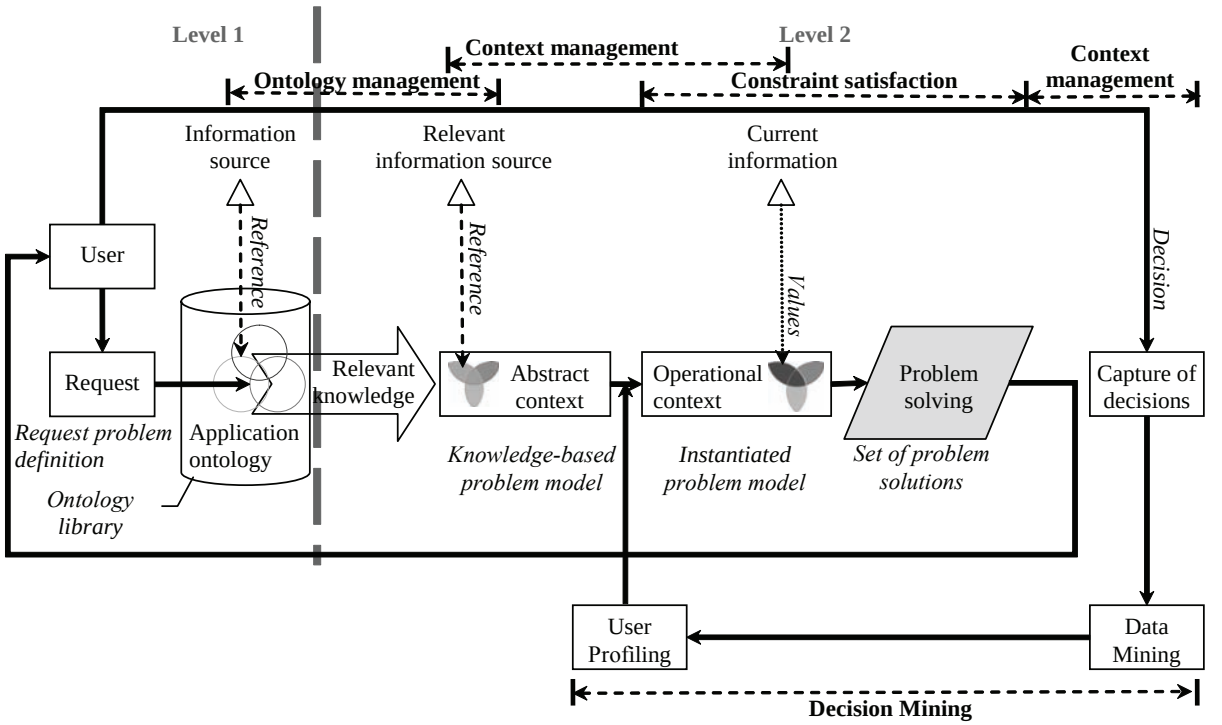
The described above conceptual framework is presented in Figure 2.

Three main forms of constraint satisfaction problems (CSP) are distinguished (Bowen, 2002):

- **The Decision CSP.** Given a network, decide whether the solution set of the network is nonempty;
- **The Exemplification CSP.** Given a network, return some tuple from the solution set if the solution set is nonempty, or return an empty solution otherwise;
- **The Enumeration CSP.** Given a network, return the solution set.

If all the domains in an operational context have been redefined the operational context is considered as a situation description. This case corresponds to CSP of the 1st type. If an operational context contains not

Figure 2. Conceptual framework of context-driven decision making support



redefined domains, it means that a solution is expected. In this case CSP falls under the 2nd or the 3rd form.

The approach proposes usage of a profile model that supports user clustering using text mining techniques (Smirnov et al., 2005b). The result of user clustering is identification of roles for decision maker grouping. Grouping shows similarities between different users and allows finding alternative decision makers with similar competence, knowledge, etc. Inputs for clustering algorithm are user requests and the ontology.

In the implemented prototype only specific action of users are considered for clustering. The list of action types can be extended. In this case different actions can be assigned different weights for similarity measurement or several classifications based on different actions can be performed. Initially a new user has an empty profile and cannot be classified. Later, when the profile is filled up with some accumulated information the user can be clustered.

To take into account other data stored in user profiles a method based on the analysis of alternatives has been developed. Unlike the mentioned above method searching for similarities of the decisions selected by the user, this method assumes an analysis of differences between decisions from the remaining solutions. These differences are stored and appropriate statistics is accumulated. Based on the statistics the user preferences can be identified as most common differences of his/her decisions from the remaining solutions taking into account information from the operational context.

Let $DM = \{Sol_{1, \ldots, n}, Dec\}$, where Sol_i is a solution and *Dec* is a decision selected by the user. The decision *Dec* is considered to be excluded from the set of solutions $Sol_{1, \ldots, n}$.

Each decision as well as the solution consists of objects (class instances) *I* with instantiated attributes *P* (attribute values *V*): $Sol_i = \{<I, P, V>\}$.

Comparison of the decision with each solution results in sets:

$Diff_i^+ = \{<I, P, V>\}$, $i = 1, \ldots, n$, containing objects from the decision and object attributes that differ from those of the solution ($\forall <I, P, V> \subset Diff_i^+$: $<I, P, V> \subset Dec$, $<I, P, V> \not\subset Sol_i$).

$Diff_i^- = \{<I, P, V>\}$, $i = 1, \ldots, n$, containing objects from the solution and object attributes that differ from

those of the decision ($\forall$ <*I, P, V*> $\subset Diff_i^-$: <*I, P, V*> $\not\subset$ *Dec*, <*I, P, V*> $\subset Sol_i$).

Then, these sets are united and number of occurrences (*k*) of each tuple <*I, P, V*> is calculated:

$Diff^+ = \{<I, P, V>, k_l\}$ – objects and their values preferred by the user.

$Diff^- = \{<I, P, V>, k_m\}$ – objects and their values avoided by the user.

When the volume of the accumulated statistics is enough (it depends on the problem complexity and is defined by the system administrator) a conclusion can be made about usually preferred and/or avoided objects and their parameters from the problem domain by the user. This conclusion is considered as the user preference.

The resulting preferences can be described via rules, e.g.:

if Weather.WindSpeed > 10 **then** EXCLUDE HELICOPTERS

if Disaster.Type = "Fire" **then** INCLUDE FIREFIGHTERS

if Disaster.Number_Of_Victims > 0 **then** Solution. Time -> min

When users are clustered into roles and their preferences are revealed it is possible to compare preferred objects in each cluster and build hypothesis what decision trees have to look like. The extracted preferences are used for building decision trees allowing making a right decision in a critical situation (semi)automatically. Also such decision trees could be useful for supporting decision maker in solution estimation and ranking. Today, there are a lot of algorithms and tools implementing them to solve this task (e.g., CART – Classification and Regression Tree algorithm (Breiman et al., 1984), C4.5 (Ross Quinlan, 1993) and others).

A detailed description of the approach and some comparisons can be found in (Smirnov et al., 2007)

Future Trends

Usually, complex situations involve a great number of different heterogeneous teams (sometimes multinational), which have to collaborate in order to succeed. Besides, it might be often necessary to use external sources to get required information. Such actions require intensive information exchange in order to achieve necessary level of the situational awareness, creation of ad-hoc action plans, and continuously updated information.

Centralized control is not always possible due to probable damages in local infrastructure, different subordination of participating teams, etc. Another disadvantage of the centralized control is its possible failure that would cause failure the entire operation. Possible solution for this is organization of a decentralized self-organizing coalitions consisting of the operation participants. However, in order keep this coalition-based network operating it is necessary to solve a number of problems that can be divided into technical (hardware-related) and methodological constituents. Some of the problems that are currently under study include providing for semantic interoperability between the participants and definition of the standards and protocols to be used. For this purpose such technologies as situation management, ontology management, profiling and intelligent agents can be used. Standards of information exchange (e.g., Web-service standards), negotiation protocols, decision making rules, etc. can be used for information / knowledge exchange and rapid establishing of ad-hoc partnerships and agreements between the participating parties.

CONCLUSION

The article presents results of a research related to the area of data mining in the context-sensitive intelligent decision support. In the proposed approach auxiliary data are saved in the user profile after each decision making process. These data are ontology elements describing actual situation, preferences of the user (decision maker), set of alternative solutions and the final decision. A method for role-based decision mining for revealing the user preferences influencing upon the final user decision has been proposed.

In the future other methods from the area of data mining can be developed or adapted to extract new results from the accumulated data.

REFERENCES

Bowen, J. (2002) Constraint Processing Offers Improved Expressiveness and Inference for Interactive Expert Systems, in *Proceedings of the International Workshop on Constraint Solving and Constraint Logic Programming' 2002*, 93-108.

Breiman, L., Friedman J. H., Olshen R.A., Stone C. T. (1984). Classification and Regression Trees, Wadsworth.

Brown, D., Liu, H., & Xue, Y. (2006). Spatial Analysis with Preference Specification of Latent Decision Makers for Criminal Event Prediction, *Special Issue of Intelligence and Security Informatics*, 560-573.

Chiang, W., Zhang, D., & Zhou, L. (2006). Predicting and Explaining Patronage Behavior Towards Web and Traditional Stores Using Neural Networks: a Comparative Analysis with Logistic Regression, *Decision Support Systems*, 41(2), 514-531.

Decision Mining software corporate Web site (2007), URL: http://www.decisionmining.com.

Dey, A. K., Salber, D., & Abowd, G. D. (2001). A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications, Context-Aware Computing, in *A Special Triple Issue of Human-Computer Interaction*, Moran , T.P., Dourish P. (eds.), Lawrence-Erlbaum, 16, 97-166.

FIPA 98 Specification (1998). Part 12 - Ontology Service. Geneva, Switzerland, Foundation for Intelligent Physical Agents (FIPA), URL: http://www.fipa.org.

Li, X. (2005). A Scalable Decision Tree System and Its Application in Pattern Recognition and Intrusion Detection, *Decision Support Systems*, 41(1), 112-130.

Ross Quinlan, J. (1993). C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers Inc.

Rozinat, A., & van der Aalst, W.M.P. (2006). Decision Mining in ProM, in *Lecture Notes in Computer Science. Proceedings of the International Conference on Business Process Management (BPM 2006)*, Dustdar, S., Fiadeiro, J., Sheth, A. (eds.), 4102, Springer-Verlag, Berlin, 420-425.

Simon, H. A. (1965) The Shape of Automation. New York: Harper & Row.

Smirnov, A., Pashkin, M., Chilov, N., & Levashova, T. (2005). Constraint-driven methodology for context-based decision support, in *Design, Building and Evaluation of Intelligent DMSS (Journal of Decision Systems)*, Lavoisier, 14(3), 279-301.

Smirnov, A., Pashkin, M., Chilov, N., Levashova, T., Krizhanovsky, A., & Kashevnik, A. (2005). Ontology-Based Users and Requests Clustering in Customer Service Management System, in *Autonomous Intelligent Systems: Agents and Data Mining: International Workshop, AIS-ADM 2005*, Gorodetsky, V., Liu, J., Skormin, V. (eds.), Lecture Notes in Computer Science, 3505, Springer-Verlag GmbH, 231-246.

Smirnov, A., Pashkin, M., Levashova, T., Shilov, N., Kashevnik, A. (2007). Role-Based Decision Mining for Multiagent Emergency Response Management, in *Proceedings of the second International Workshop on Autonomous Intelligent Systems*, Gorodetsky, V., Zhang, C., Skormin, V., Cao, L. (eds.), LNAI 4476, Springer-Verlag, Berlin, Heidelberg, 178-191.

Thomassey, S., & Fiordaliso, A. (2006). A Hybrid Sales Forecasting System Based on Clustering and Decision Trees, *Decision Support Systems*, 42(1), 408-421.

W3C (2007). OWL Web Ontology Language Overview, URL: http://www.w3.org/TR/owl-features/.

KEY TERMS

Abstract Context: An ontology-based model integrating information and knowledge relevant to the problem at hand.

Context: Any information that can be used to characterize the situation of an entity where an entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves (Dey et al., 2001).

Decision Mining: A data mining area aimed at finding "rules" explaining under what circumstances one activity is to be selected rather than the other one.

Object-Oriented Constraint Network Formalism (OOCN): An ontology representation formalism, according to which an ontology (*A*) is defined as: *A* = (*O, Q, D, C*) where: *O* is a set of *object classes* ("*classes*"), *Q* is a set of class attributes ("*attributes*"), *D* is a set of attribute domains ("*domains*"), and *C* is a set of *constraints*.

Ontology: An *explicit specification* of the structure of the certain *domain* that includes *a vocabulary* (i.e. a list of logical constants and predicate symbols) for referring to the subject area, and *a set of* logical state-

ments expressing the *constraints* existing *in the domain* and restricting the interpretation of the vocabulary; it provides a vocabulary for representing and communicating *knowledge* about some topic and *a set of relationships and properties* that hold for the *entities* denoted by that vocabulary (FIPA, 1998).

Operational Context: An instantiation of the abstract context with data provided by the information sources or calculated based on functions specified in the abstract context.

Web Ontology Language (OWL): The OWL Web Ontology Language is designed to be used by applications that need to process the content of information instead of just presenting information to humans. OWL facilitates greater machine interpretability of Web content than that supported by XML, RDF, and RDF Schema (RDF-S) by providing additional vocabulary along with a formal semantics. OWL has three increasingly-expressive sublanguages: OWL Lite, OWL DL, and OWL Full (W3C, 2007).

Context-Sensitive Attribute Evaluation

Marko Robnik-Šikonja
University of Ljubljana, FRI

INTRODUCTION

The research in machine learning, data mining, and statistics has provided a number of methods that estimate the usefulness of an attribute (feature) for prediction of the target variable. The estimates of attributes' utility are subsequently used in various important tasks, e.g., feature subset selection, feature weighting, feature ranking, feature construction, data transformation, decision and regression tree building, data discretization, visualization, and comprehension. These tasks frequently occur in data mining, robotics, and in the construction of intelligent systems in general.

A majority of attribute evaluation measures used are myopic in a sense that they estimate the quality of one feature independently of the context of other features. In problems which possibly involve much feature interactions these measures are not appropriate. The measures which are historically based on the Relief algorithm (Kira & Rendell, 1992) take context into account through distance between the instances and are efficient in problems with strong dependencies between attributes.

BACKGROUND

The majority of feature evaluation measures are impurity based, meaning that they measure impurity of the class value distribution. These measures evaluate each feature separately by measuring impurity of the splits resulting from partition of the learning instances according to the values of the evaluated feature. Assuming the conditional independence of the features upon the class, these measures are myopic, as they do not take the context of other features into account. If the target concept is a discrete variable (the classification problem) well-known and used measures of these kind are information gain (Hunt et al., 1966), Gini index (Breiman et al., 1984), j-measure (Smyth & Goodman, 1990), Gain ration (Quinlan, 1993) and MDL (Kononenko, 1995). Large differences in the impurity of class values before the split, and after the split on a given feature, imply purer splits and therefore more useful features. We cannot directly apply these measures to numerical features, but we can use discretization techniques and then evaluate discretized features. If the target concept is presented as a real valued function (regression problem), the impurity based evaluation heuristics used are e.g., the mean squared and the mean absolute error (Breiman et al., 1984).

The term context here represents related features, which interact and only together contain sufficient information for classification of instances. Note that the relevant context may not be the same for all instances in a given problem. The measures which take the context into account through distance between the instances and are efficient in classification problems with strong dependencies between attributes are Relief (Kira & Rendell, 1992), Contextual Merit (Hong, 1997), and ReliefF (Robnik-Sikonja & Kononenko, 2003). RReliefF is a measure proposed to address regression problems (Robnik-Sikonja & Kononenko, 2003).

For a more thorough overview of feature quality evaluation measures see (Kononenko & Kukar, 2007). Breiman (2001) has proposed random forest learning algorithm which, as a byproduct, can output the utility of the attributes. With large enough data sample which ensures sufficiently large and diverse trees in the forest these estimates are also context-sensitive. For an overview of other recent work, especially in the context of feature subset selection see (Guyon & Elisseeff, 2003). Note that this chapter is solely a machine learning view of feature selection and omits methods for model selection in regression that amount to feature selection. A recent work trying to bridge the two worlds is (Zhou et al., 2006).

MAIN FOCUS

The main idea of how to take the context of other features into account was first presented in algorithm Relief (Kira & Rendell, 1992), which is designed for two-class problems without missing values. The idea of the algorithm, when analyzing learning instances, is to take into account not only the difference in feature values and the difference in classes, but also the distance between the instances. Distance is calculated in the feature space, therefore similar instances are close to each other and dissimilar are far apart. By taking the similarity of instances into account, the context of all the features is implicitly considered.

The algorithm Relief illustrated in Box 1 randomly selects an instance and finds the nearest instance from the same class (nearest hit) and the nearest instance from the opposite class (nearest miss). Then it updates the quality of each feature with respect to whether the feature differentiates two instances from the same class (undesired property of the feature) and whether it differentiates two instances from opposite classes (desired property). By doing so, the quality estimate takes into account the local ability of the feature to differentiate between the classes. Repeating the whole procedure for large enough sample these local estimates provide a global picture of the feature utility, but the locality implicitly takes into account the context of other features.

Let's say that a given feature explains the change of the class value of the instance, when the change of its values is one of the minimal changes required for changing the class value. The quality evaluations of Relief algorithms can then be interpreted as the portions of the explained concept i.e., as the ratio between the number of the explained changes in the concept and the number of examined instances.

ReliefF for Classification and RReliefF for Regression

A more realistic and practically useful variant of Relief is its extensions, called ReliefF for classification and RReliefF for regression problems (Robnik-Sikonja & Kononenko, 2003). Unlike original Relief these two algorithms are able to deal with incomplete and noisy data. The most important difference is in searching for the nearest hit and miss. Noise or mistakes in class and/or feature values significantly affects the selection of nearest hits and misses. In order to make this process more reliable in the presence of noise, ReliefF and RReliefF use several nearest hits and misses and average their contributions to features' quality estimates. ReliefF can be used also for evaluating the feature quality in multi-class problems and to do so it searches for nearest instances from each class. The contributions of different classes are weighted with their prior probabilities. In regression problems the target variable is numerical, therefore nearest hits and misses cannot be used in a strict sense. RReliefF (Regressional ReliefF) uses a kind of ``probability'' that two instances belong to two ``different'' classes. This ``probability'' is modeled with the distance between the values of the target variable of two learning instances.

Box 1.

Algorithm Relief
Input: set of instances $<x_i, \tau_i>$
Output: the vector W of attributes' evaluations

```
set all weights W[A] := 0.0;
for i := 1 to #sample_size do begin
  randomly select an instance R;
  find nearest hit H and nearest miss M;
  for A := 1 to #all_attributes do
   W[A] := W[A] - diff(A,R,H)/m + diff(A,R,M)/m;
end;
```

Extensions of ReliefF

The ideas of context-sensitive attribute evaluation introduced in Relief, ReliefF and RReliefF have been adapted for efficient feature evaluation in other areas like inductive logic programming, cost-sensitive learning, and evaluation of ordered features at value level (Kononenko & Robnik-Sikonja, 2007).

In inductive logic programming aka multi-relational data mining (Dzeroski & Lavrac, 2001), the goal of learning is to develop predicate descriptions from examples and background knowledge. The examples, background knowledge and final descriptions are all described as logic programs. The key idea of context-sensitive feature evaluation is to estimate literals according to how well they distinguish between the instances that are logically similar. For that one has to change the notion of a distance between the literals.

In cost-sensitive classification all errors are not equally important (Elkan, 2001). In general, differences in importance of errors are handled through the cost of misclassification. Cost-sensitivity is a desired property of an algorithm which tries to rank or weight features according to their importance. The key idea in cost-sensitive ReliefF is to use the expected cost of misclassifying an instance in weighting the quality updates (Robnik-Sikonja, 2003).

A context sensitive algorithm for evaluation of ordinal features was proposed in (Robnik-Sikonja & Vanhoof, 2007) and used on a customer (dis)satisfaction problem from marketing research. The ordEval algorithm exploits the information hidden in ordering of features' and class values and provides a separate score for each value of the feature. Similarly to ReliefF the contextual information is exploited via selection of nearest instances. The ordEval outputs probabilistic factors corresponding to the effect an increase/decrease of feature's value has on the class value. This algorithm and its visualizations can be used as an exploratory tool for analysis of any survey with graded answers.

Applications

(R)ReliefF has been applied in a variety of different ways in data mining and many studies have reported its good performance in comparison with other feature evaluation methods. Besides the usual application for filter subset selection, (R)ReliefF was used for feature ranking, feature weighing, building tree-based models and associative rules, feature discretization, controlling the search in genetic algorithms, literal ranking in ILP, and constructive induction. It is implemented in many data mining tools, including CORElearn, Weka, Orange and R.

FUTURE TRENDS

The main challenge of context-sensitive feature evaluation is handling ever-increasing number of instances and features. When dealing with data sets with a huge number of instances feature selection methods typically perform a random sampling. Liu & Motoda (2004) introduce the concept of active feature selection, and apply selective sampling based on data variance to ReliefF. They reduced the required number of training instances and achieve considerable time savings without performance deterioration. Context-sensitive algorithms are especially sensitive to large number of features as they blur the distance and the context. In high-dimensional spaces the performance may deteriorate and we need tools for analysis and detection of such behavior. The stability of feature selection algorithms was studied by (Kalousis et al., 2007). One possible solution is to make the feature evaluation procedure iterative. For ReliefF such an attempt was made by (Draper et al., 2003). Relief algorithms are computationally more complex than some other (myopic) feature estimation measures. However, they also have a possible advantage that they can be naturally split into several independent tasks which is a prerequisite for successful parallelization of an algorithm. Iterations in the algorithm are natural candidates for separate processes, which would turn Relief into the fine-grained parallel algorithm. With arrival of microprocessors with multiple cores this will be an easy speedup.

Feature evaluation is a lively research topic in data mining, mostly motivated by the feature subset selection challenges posed by image analysis, web applications, and bioinformatics, in particular gene expression analysis (Gadat & Younes, 2007; Ooi et al., 2007; Nilsson et al., 2007). New approaches and discoveries are also in the basic approach towards feature evaluation. Torkkola (2003) has proposed a feature selection method based on non-parametric mutual information maximization. While this approach is useful for sets of features, its estimates could be adapted to serve for evaluation of individual features without independence assump-

tion. Zhao & Liu (2007) propose a unifying approach to feature evaluation in supervised and unsupervised learning based on pairwise distance similarity graph and show that ReliefF is a special case of the proposed framework.

CONCLUSION

ReliefF in classification and RReliefF in regression exploit the context of other features through distance measures and can detect highly conditionally dependent features. The basic idea of these algorithms is to evaluate features in the local context. These algorithms are able to deal with the incomplete data, with multi-class problems, and are robust with respect to the noise. Various extensions of the ReliefF algorithm, like evaluation of literals in inductive logic programming, cost-sensitive feature evaluation with ReliefF, and the ordEval algorithm for the evaluation of features with ordered values, show the general applicability of the basic idea. (R)ReliefF family of algorithms has been used in many different data mining (sub)problems and applications. The comprehensive interpretability of the (R)ReliefF's estimates makes this approach attractive also in the application areas where the knowledge discovery is more important than predictive performance, which is often case in the medicine and social sciences. In spite of some claims that (R)ReliefF is computationally demanding, this is not an inherent property and efficient implementations and use do exist.

REFERENCES

Breiman, L. & Friedman, J. H. & Olshen, R. A. & Stone, C. J. (1984). *Classification and regression trees*. *Wadsworth Inc.*

Breiman, L. (2001). Random Forests. *Machine Learning Journal, 45*, 5-32.

Dzeroski, S. & Lavrac, N. (Eds.) (2001). *Relational Data Mining*. Springer, Berlin.

Draper, B. & Kaito, C. & Bins, J. (2003). Iterative Relief. *Workshop on Learning in Computer Vision and Pattern Recognition, Madison, WI.*

Elkan, C. (2001). The Foundations of Cost-Sensitive Learning. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI'01).*

Gadat, S. & Younes, L. (2007). A Stochastic Algorithm for Feature Selection in Pattern Recognition. *Journal of Machine Learning Research* 8(Mar):509-547.

Guyon, I. & Elisseeff, A. (2003). An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research, 3*, 1157-1182.

Hong, S. J. (1997). Use of Contextual Information for Feature Ranking and Discretization. *IEEE Transactions on Knowledge and Data Engineering, 9*, 718-730.

Hunt, E. B. & Martin, J. & Stone, P. J. (1966). *Experiments in Induction. Academic Press.*

Kira, K. & Rendell, L. A. (1992). A practical approach to feature selection. In Sleeman, D. & Edwards, P. (Eds.) *Machine Learning: Proceedings of International Conference (ICML'92), Morgan Kaufmann, San Francisco*, 249-256.

Kononenko, I. (1995). On Biases in Estimating Multi-Valued Attributes. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'95)*. Morgan Kaufmann, 1034-1040.

Kononenko, I. & Kukar, M. (2007). *Machine Learning and Data Mining, Introduction to Principles and Algorithms. Horwood Publishing, Chichester.*

Kononenko, I. & Robnik-Šikonja, M. (2007). Non-myopic feature quality evaluation with (R)ReliefF. In Liu, H. & Motoda, H. (Eds.). *Computational Methods of Feature Selection. Chapman and Hall/CRC Pres.*

Liu, H. & Motoda, H. & Yua, L. (2004). A selective sampling approach to active feature selection. *Artificial Intelligence* 159, 49–74.

Nilsson, R. & Peña, J.M. & Björkegren, J. & Tegnér, J. (2007). Consistent Feature Selection for Pattern Recognition in Polynomial Time. *Journal of Machine Learning Research*, 8(Mar):589-612.

Ooi, C.H. & Chetty, M. & Teng, S.W. (2007). Differential prioritization in feature selection and classifier aggregation for multiclass microarray datasets. *Data Mining and Knowledge Discovery*, 14(3): 329-366.

Quinlan, J. R. (1993) *C4.5: Programs for Machine Learning. Morgan Kaufmann, San Francisco.*

Robnik-Šikonja, M. (2003). Experiments with Cost-sensitive Feature Evaluation. In Lavrač, N.& Gamberger, D.& Blockeel, H. & Todorovski, L. (Eds.). *Machine Learning: Proceedings of ECML' 2003*, pp. 325-336, *Springer, Berlin.*.

Robnik-Šikonja, M. & Kononenko, I. (2003). Theoretical and Empirical Analysis of ReliefF and RReliefF. *Machine Learning Journal, 53*, 23-69.

Robnik-Šikonja, M. & Vanhoof, K. (2007). Evaluation of ordinal attributes at value level. *Data Mining and Knowledge Discovery, 14*, 225-243.

Smyth, P. & Goodman, R. M. (1990). Rule induction using information theory. In Piatetsky-Shapiro, G. & Frawley, W. J. (Eds.). *Knowledge Discovery in Databases. MIT Press.*

Torkkola, K. (2003). Feature Extraction by Non-Parametric Mutual Information Maximization. *Journal of Machine Learning Research*, 3, 1415-1438.

Zhao, Z. & Liu, H. (2007). Spectral feature selection for supervised and unsupervised learning. In Ghahramani, Z. (Ed.) *Proceedings of the 24th international Conference on Machine Learning ICML '07.*

Zhou, J. & Foster, D.P., & Stine, R.A. & Ungar, L.H. (2006). Streamwise Feature Selection. *Journal of Machine Learning Research,* 7(Sep): 1861-1885.

KEY TERMS

Attribute Evaluation: A data mining procedure which estimates the utility of attributes for given task (usually prediction). Attribute evaluation is used in many data mining tasks, for example in feature subset selection, feature weighting, feature ranking, feature construction, decision and regression tree building, data discretization, visualization, and comprehension.

Context of Attributes: In a given problem the related attributes, which interact in the description of the problem. Only together these attributes contain sufficient information for classification of instances. The relevant context may not be the same for all the instances in given problem.

Cost-Sensitive Classification: Classification where all errors are not equally important. Usually differences in importance of errors are handled through the cost of misclassification.

Evaluation of Ordered Attributes: For ordered attributes the evaluation procedure should take into account their double nature: they are nominal, but also behave as numeric attributes. So each value may have its distinct behavior, but values are also ordered and may have increasing impact.

Feature Construction: A data transformation method aimed to increase the expressive power of the models. Basic features are combined with different operators (logical and arithmetical) before or during learning.

Feature Ranking: The ordering imposed on the features to establish their increasing utility for the given data mining task.

Feature Subset Selection: Procedure for reduction of data dimensionality with a goal to select the most relevant set of features for a given task trying not to sacrifice the performance.

Feature Weighting: Under the assumption that not all attributes (dimensions) are equally important feature weighting assigns different weights to them and thereby transforms the problem space. This is used in data mining tasks where the distance between instances is explicitly taken into account.

Impurity Function: A function or procedure for evaluation of the purity of (class) distribution. The majority of these functions are coming from information theory, e.g., entropy. In attribute evaluation the impurity of class values after the partition by the given attribute is an indication of the attribute's utility.

Non-Myopic Attribute Evaluation: An attribute evaluation procedure which does not assume conditional independence of attributes but takes context into account. This allows proper evaluation of attributes which take part in strong interactions.

Ordered Attribute: An attribute with nominal, but ordered values, for example, increasing levels of satisfaction: low, medium, and high.

Control-Based Database Tuning Under Dynamic Workloads

Yi-Cheng Tu
University of South Florida, USA

Gang Ding
Olympus Communication Technology of America, Inc., USA

INTRODUCTION

Database administration (tuning) is the process of adjusting database configurations in order to accomplish desirable performance goals. This job is performed by human operators called database administrators (DBAs) who are generally well-paid, and are becoming more and more expensive with the increasing complexity and scale of modern databases. There has been considerable effort dedicated to reducing such cost (which often dominates the total ownership cost of mission-critical databases) by making database tuning more automated and transparent to users (Chaudhuri *et al*, 2004; Chaudhuri and Weikum, 2006). Research in this area seeks ways to automate the hardware deployment, physical database design, parameter configuration, and resource management in such systems. The goal is to achieve acceptable performance on the whole system level without (or with limited) human intervention.

According to Weikum *et al*. (2002), problems in this category can be stated as:

workload × configuration (?) → performance

which means that, given the features of the incoming workload to the database, we are to find the right settings for all system knobs such that the performance goals are satisfied The following two are representatives of a series of such tuning problems in different databases:

- **Problem 1: Maintenance of multi-class service-level agreements (SLA) in relational databases.** Database service providers usually offer various levels of performance guarantees to requests from different groups of customers. Fulfillment of such guarantees (SLAs) is accomplished by allocating different amounts of system resources to different queries. For example, query response time is negatively related to the amount of memory buffer assigned to that query. We need to dynamically allocate memory to individual queries such that the absolute or relative response times of queries from different users are satisfied.
- **Problem 2: Load shedding in stream databases.** Stream databases are used for processing data generated continuously from sources such as a sensor network. In streaming databases, data processing delay, i.e., the time consumed to process a data point, is the most critical performance metric (Tatbul *et al*., 2003). The ability to remain within a desired level of delay is significantly hampered under situations of overloading (caused by bursty data arrivals and time-varying unit data processing cost). When overloaded, some data is discarded (i.e., load shedding) in order to keep pace with the incoming load. The system needs to continuously adjust the amount of data to be discarded such that 1) delay is maintained under a desirable level; 2) data is not discarded unnecessarily.

Such problems can hardly be solved by using rules of thumbs and simply throwing in more hardware. In the following section, we shall also see that the traditional approach of treating tuning problems as static optimization problems does not work well for dynamic workloads such as those with OLAP queries. In this chapter, we introduce an emerging new approach to attack self-tuning database problems that is based on well-established results in feedback control theory. Specifically, we address the core issues of the approach and identify critical challenges of applying control theory in the area of self-tuning databases.

BACKGROUND

Current research in automatic tuning (or self-tuning) of databases tend to treat the problem as an optimization problem with the performance metrics and workload characteristics as inputs. The main drawback for this strategy is: real-world workloads, especially OLAP workloads, are highly unpredictable in that their parameters and behaviors can change very frequently (Tu *et al*., 2005). Such uncertainties in workloads can bring dramatic variations to system performance and cause the database to run in suboptimal status. In order to maintain consistently good performance, we need to develop means for the database to quickly adapt to the changes in workload.

One way to address the above challenge is to treat the problem as an online optimization (Chaudhuri and Weikum, 2006) and solve it by incremental algorithms. However, there is generally no guarantee on the accuracy and convergence of such algorithms, and some problems have no incremental solutions. Another important question, which is either ignored or answered empirically in current studies, is *how often do we need to rerun the optimization*? Our observation is that people tend to follow *ad hoc* strategies for individual problems in this field. It would be desirable to have a common theoretical framework under which a series of problems can be approached.

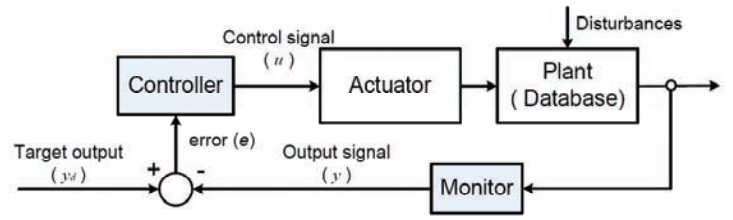
In this chapter, we argue that control theory provides such a foundation to approach the aforementioned problems in self-tuning databases. The reason for this is: designing systems with resistance to internal/external uncertainties is one of the main goals of control theory (Hellerstein *et al*., 2004). Note that control theory is not a single technique. Instead, it is the collection of a rich set of mathematical tools for analyzing system dynamics and designing mechanisms with guaranteed performance. We discuss some of the core issues of using control techniques in self-tuning databases. Currently, we have seen control-theoretical methods in solving database tuning problems and the effectiveness of the method is supported by both analytical and experimental results (Lightstone *et al*., 2007; Storm *et al*., 2006; Tu *et al*., 2005, 2006, 2007; Tu and Prabhakar 2006). For example, a load shedding framework based on feedback control designed by Tu and coworkers (2006, 2007) achieves processing delay violations that are 2 to 3 orders of magnitude lower (with the same amount of data loss), as compared to optimization-based methods. Storm *et al.*, (2006) reported the implementation of the Self-Tuning Memory Manager (STMM) with a control-based design and up to 254% increase in performance (e.g., decrease in query response time) in DB2 Version 9.1.

MAIN FOCUS

In this chapter, the term *control* refers to the manipulation of particular feature(s) (i.e., output signal) of a system (called *plant* in control terminology) by adjusting inputs injected into it (Hellerstein *et al*., 2004). We focus on *feedback control* where output signals are taken into account in making control decisions. The main components of a feedback control system form a *feedback control loop* (Figure 1): a *monitor* measures the output signal y of the plant; the measurements are compared with a desirable output value y_d and the difference between them is called *control error*; a *controller* maps control error e to a control signal u; an *actuator* adjusts the behavior of the plant according to signal u. The goal of the control operations is to overcome the effects of system and environmental uncertainties (named *disturbance*s) such that the output signal tracks the target value.

The above conceptual design can be mapped into a concrete model for our problem: the plant is the database system; the actuator is the existing database mechanisms

Figure 1. Components of a feedback control loop



to implement a tuning decision; output signal is the performance metric(s) we consider; and the control signal is the database configurations we need to tune to. For example, in Problem 1, we have the query response time as output and resource allocation to multiples queries as input. Clearly, the most critical part of the control loop is the controller. Control theory is the mathematical foundation to the design of controllers based on the dynamic features of the plant. Carefully designed controllers can overcome unpredictable disturbances with guaranteed runtime performance such as *stability* and *convergence time*. However, due to the inherent differences between database systems and traditional control systems (i.e., mechanical, electrical, chemical systems), the application of control theory in database systems is by no means straightforward.

Modeling Database Systems

Rigorous control theory is built upon the understanding of the dynamics of the system to be controlled. Derivation of models that describe such dynamics is thus a critical step in control engineering. Therefore, the first challenge of control-based tuning is to *generate dynamic models of various database systems*. Linear systems have been well-studied in the control community. Generally, the dynamic behavior of a single-input-single-output (SISO) linear time-invariant (LTI) system can be modeled by a transfer function between the input u and output y in the frequency domain:

$$G(s) = \frac{a_n s^{n-1} + a_{n-1} s^{n-2} + ... + a_1}{s^n + b_n s^{n-1} + b_{n-1} s^{n-2} + ... + b_1}$$

where n is called the *order* of the model. The above transfer function only gives the relationship between the input and output. All the underlying n system dynamics x can be represented by a state-space model in time domain:

$$\begin{cases} x = Ax + Bu \\ y = Cx + Du \end{cases}$$

where A, B, C, and D are model parameters. Given an accurate model, all dynamics of the object can be analytically obtained, based on which we can analyze important characteristics of the output and states such as stability of the output, the observability, and controllability of each state.

Depending on the available information about the system, there are different ways to obtain the model. When the physical mechanism that drives the system is clearly known, the structure and all parameters of the model can be accurately derived. However, for complex systems such as a database, the analytical model may be difficult to derive. Fortunately, various *system identification* techniques (Franklin *et al*., 2002; Tu *et al*., 2005) can be used to generate approximate models for databases. The idea is to feed the database system with inputs with various patterns. By comparing the actual outputs measured with the derivation of the output as a function of unknown parameters and input frequency, the model order n and parameters a_i, $b_i (1 \le i \le n)$ can be calculated.

Control System Design

The dynamic model of a system only shows the relationship between system input and output. The second challenge of control-based tuning is to *design feedback control loops that guide us how to change the input* (*database configuration*) *in order to achieve desirable output* (*performance*). We need to test various design techniques and identify the ones that are appropriate for solving specific database tuning problems. When the linear system model is known, the proportional-integral-derivative (PID) controller is commonly used. It has three components and we can use mathematical tools such as *root locus* and *bode plot* to find controller parameters that meet our runtime performance requirements (Franklin *et al*., 2002). When model is partly known, more complicated design methods should be involved. For example, when some parameters of the model are unknown or time-variant, the controller can only be designed for an estimated model by online system identification (i.e., adaptive control). For systems with unstructured noises which vary fast, robust control has been actively studied (Ioannou and Datta, 1991).

Database-Specific Issues

Although control theory provides a sound theoretical background for the aforementioned problems, its application in self-tuning databases is non-trivial. The inherent differences between database/information systems and traditional control systems (i.e., mechanical, electrical, chemical systems) bring additional chal-

lenges to the design of the control loop. In this section, we discuss some of the challenges we identified:

1. **Lack of real-time output measurement:** Response time and processing delay are important performance metrics in database systems. Accurate measurements of system output in real-time is essential in traditional control system design. Unfortunately, this requirement is not met in self-tuning database problems where response time and processing delays are the output signals. Under this situation, the output measurement is not only delayed, but also delayed by an unknown amount (the amount is the output itself!). A solution to this is to estimate the output from measurable signals such as queue length.
2. **Actuator design:** In traditional control systems, the actuator can precisely apply the control signal to the plant. For example, in the cruise control system of automobiles, the amount of gasoline injected into the engine can be made very close to the value given by the controller. However, in database systems, we are not always sure that the control signal given by the controller can be correctly generated by our actuator. Two scenarios that cause the above difficulties in actuator design are: 1) Sometimes the control signal is implemented as a modification of the original (uncontrolled) system input signal that is unpredictable beforehand; 2) Another source of errors for the control signal is caused by the fact that the actuator is implemented as a combination of multiple knobs.
3. **Determination of the control period:** The control (sampling) period is an important parameter in digital control systems. An improperly selected sampling period can deteriorate the performance of the feedback control loop. As we mentioned earlier, current work in self-tuning databases consider the choice of control period as an empirical practice. Although the right choice of control period should always be reasoned case by case, there are certain theoretical rules we can follow. For controlling database systems, we consider the following issues in selecting the control period: 1) *Nature of disturbances*. In order to deal with disturbances, the control loop should be able to capture the moving trends of these disturbances. The basic guiding rule for this is the Nyquist-Shannon sampling theorem; 2) *Uncertainties in system signals*. Computing systems are inherently discrete. In database problems, we often use some statistical measurements of continuous events occurring within a control period as output signal and/or system parameter. This requires special consideration in choosing the control period; and 3) *Processing (CPU) overhead*. High sampling frequency would interfere with the normal processing of the database and causes extra delays that are not included in the model. In practice, the final choice of control period is the result of a tradeoff among all the above factors.
4. **Nonlinear system control:** Non-linear characteristics are common in database systems. When the model is nonlinear, there is no generic approach to analyze or identify the model. The most common approach is to linearize the model part by part, and analyze or identify each linear part separately. For the worst case when no internal information about the system is available, there are still several techniques to model the object. For example, the input-output relationship can be approximated by a set of rules provided by people familiar with the system, and a rule is represented by a mapping from a fuzzy input variable to a fuzzy output variable. An artificial neural network model can also be employed to approximate a nonlinear function. It has been proven that a well-tuned fuzzy or neural network model can approximate any smooth nonlinear function within any error bound (Wang *et al*., 1994).

FUTURE TRENDS

Modeling database systems is a non-trivial task and may bear different challenges in dealing with different systems. Although system identification had provided useful models, a deeper understanding of the underlying dynamics that are common in all database systems would greatly help the modeling process. We expect to see research towards a fundamental principle to describe the dynamical behavior of database systems, as Newton's Law to physical systems.

Another trend is to address the database-specific challenges in applying control theory to database tuning problems from a control theoretical viewpoint. In our work (Tu *et al*., 2006), we have proposed solutions to

such challenges that are specific to the context in which we define the tuning problems. Systematic solutions that extend current control theory are being developed (Lightstone *et al*., 2007). More specifically, tuning strategies based on fuzzy control and neural network models are just over the horizon.

CONCLUSION

In this chapter, we introduce the new research direction of using feedback control theory for problems in the field of self-tuning databases under a dynamic workload. Via concrete examples and accomplished work, we show that various control techniques can be used to model and solve different problems in this field. However, there are some major challenges in applying control theory to self-tuning database problems and addressing such challenges is expected to become the main objectives in future research efforts in this direction. Thus, explorations in this direction can not only provide a better solution to the problem of database tuning, but also many opportunities to conduct synergistic research between the database and control engineering communities to extend our knowledge in both fields.

REFERENCES

Chaudhuri, S., Dageville, P., & Lohman, G. M. (2004). Self-managing technology in database management systems. *Proceedings of 30th Very Large Data Bases Conference* (pp.1243-1243).

Chaudhuri, S., & Weikum, G. (2006). Foundations of automated database tuning. *Proceedings of International Conference of Date Engineering* (pp. 104).

Franklin, G. F., Powell, J. D. & Emami-Naeini, A. (2002). *Feedback control of dynamic systems*. Massachusetts: Prentice Hall.

Hellerstein, J. L., Diao, Y., Parekh, S., & Tilbury, D. (2004). *Feedback control of computing systems*. Wiley-Interscience.

Ioannou, P. A. & Datta, A. (1991). Robust adaptive control: A unified approach. *Proceedings of IEEE, 79*(12), 1736-1768.

Lightstone, S., Surendra, M., Diao, Y., Parekh, S., Hellerstein, J., Rose, K., Storm, A., & Garcia-Arellano, C. (2007). Control theory: A foundational technique for self managing databases. *Proceedings of 2nd International Workshop on Self-Managing Database Systems* (pp.395-403).

Paxson, V., & Floyd, S. (1995). Wide-area traffic: The failure of poisson modeling. *IEEE/ACM Transactions on Networking*, *3*(3), 226-244.

Storm, A., Garcia-Arellano, C., Lightstone, S., Diao, Y., & Surendra, M. (2006). Adaptive self-tuning memory in DB2. *Proceedings of 32nd Very Large Data Bases Conference* (pp. 1081-1092).

Tatbul, N., Cetintemel, U., Zdonik, S., Cherniack, M., & Stonebraker, M. (2003). Load shedding in a data stream manager. *Proceedings of 29th Very Large Data Bases Conference* (pp.309-320).

Tu, Y., Liu, S., Prabhakar, S., & Yao, B. (2006). Load shedding in stream databases: A control-based approach. *Proceedings of 32nd Very Large Data Bases Conference* (pp. 787-798).

Tu, Y., Liu, S., Prabhakar, S., Yao, B., & Schroeder, W. (2007). Using control theory for load shedding in data stream management. *Proceedings of International Conference of Data Engineering* (pp.1491-1492).

Tu, Y., Hefeeda, M., Xia, Y., Prabhakar, S., & Liu, S. (2005). Control-based quality adaptation in data stream management systems. *Proceedings of International Conference of Database and Expert Systems Applications* (pp.746-755).

Tu, Y., & Prabhakar, S. (2006). Control-based load shedding in data stream management systems. *Proceedings of International Conference on Data Engineering* (pp.144-144), PhD workshop/symposium in conjunction with ICDE 2006.

Wang, L. X. (1994). *Adaptive fuzzy systems and control: Design and stability analysis*. New Jersey, Prentice Hall.

Weikum, G., Moenkeberg, A., Hasse, C. & Zabback, P. (2002). Self-tuning database technology and information services: From wishful thinking to viable engineering. *Proceedings of 28th Very Large Data Bases Conference* (pp. 20-31).

KEY TERMS

Control Theory: The mathematical theory and engineering practice of dealing with the behavior of dynamical systems by manipulating the inputs to a system such that the outputs of the system follow a desirable reference value over time.

Controllability and Observability: Controllability of a state means that the state can be controlled from any initial value to any final value within finite time by only the inputs. Observability of a state means that, for any possible sequence of state and control inputs, the current state can be determined in finite time using only the outputs.

Convergence Rate: Another evaluation metric for closed-loop performance. It is defined as the time needed for system to bring the system output back to the desirable value in response to disturbances.

Feedback Control: the control method that takes output signals into account in making control decisions. Feedback control is also called closed-loop control due to the existence of the feedback control loop. On the contrary, the control that does not use output in generating control signal is called open-loop control.

Nyquist-Shannon Sampling Theorem: A fundamental principle in the field of information theory, the theorem states that: when sampling a signal, the sampling frequency must be greater than twice the signal frequency in order to reconstruct the original signal perfectly from the sampled version. In practice, a sampling frequency that is one order of magnitude larger than the input signal frequency is often used. The Nyquist-Shannon sampling theorem is the basis for determining the control period in discrete control systems.

System Identification: The process of generating mathematical models that describe the dynamic behavior of a system from measured data. System identification has evolved into an active branch of control theory due to the difficulty of modeling complex systems using analytical methods (so-called white-box modeling).

System Stability: One of the main features to consider in designing control systems. Specifically, stability means that bounded input can only give rise to bounded output.

Workload: A collection of tasks with different features such as resource requirements and frequencies. In the database field, a workload generally consists of various groups of queries that hold different patterns of data accessing, arrival rates (popularity), and resource consumption. Representative workloads are encapsulated into well-known database benchmarks.

Cost-Sensitive Learning

Victor S. Sheng
New York University, USA

Charles X. Ling
The University of Western Ontario, Canada

INTRODUCTION

Classification is the most important task in inductive learning and machine learning. A classifier can be trained from a set of training examples with class labels, and can be used to predict the class labels of new examples. The class label is usually discrete and finite. Many effective classification algorithms have been developed, such as naïve Bayes, decision trees, neural networks, and so on. However, most original classification algorithms pursue to minimize the error rate: the percentage of the incorrect prediction of class labels. They ignore the difference between types of misclassification errors. In particular, they implicitly assume that all misclassification errors cost equally.

In many real-world applications, this assumption is not true. The differences between different misclassification errors can be quite large. For example, in medical diagnosis of a certain cancer, if the cancer is regarded as the positive class, and non-cancer (healthy) as negative, then missing a cancer (the patient is actually positive but is classified as negative; thus it is also called "false negative") is much more serious (thus expensive) than the false-positive error. The patient could lose his/her life because of the delay in the correct diagnosis and treatment. Similarly, if carrying a bomb is positive, then it is much more expensive to miss a terrorist who carries a bomb to a flight than searching an innocent person.

BACKGROUND

Cost-sensitive learning takes costs, such as the misclassification cost, into consideration. It is one of the most active and important research areas in machine learning, and it plays an important role in real-world data mining applications. A comprehensive survey (Turney, 2000) lists a large variety of different types of costs in data mining and machine learning, including misclassification costs, data acquisition cost (instance costs and attribute costs), active learning costs, computation cost, human-computer interaction cost, and so on. The misclassification cost is singled out as the most important cost, and it has also been mostly studied in recent years (e.g., (Domingos, 1999; Elkan, 2001; Zadrozny & Elkan, 2001; Zadrozny et al., 2003; Ting 1998; Drummond & Holte, 2000, 2003; Turney, 1995; Ling et al, 2004, 2006b; Chai et al., 2004; Sheng & Ling, 2006)).

Broadly speaking, cost-sensitive learning can be categorized into two categories. The first one is to design classifiers that are cost-sensitive in themselves. We call them the direct method. Examples of direct cost-sensitive learning are ICET (Turney, 1995) and cost-sensitive decision tree (Drummond & Holte, 2000, 2003; Ling et al, 2004, 2006b). The other category is to design a "wrapper" that converts any existing cost-insensitive (or cost-blind) classifiers into cost-sensitive ones. The wrapper method is also called cost-sensitive meta-learning method, and it can be further categorized into thresholding and sampling. Here is a hierarchy of the cost-sensitive learning and some typical methods. This paper will focus on cost-sensitive meta-learning that considers the misclassification cost only.

Cost-sensitive learning:

- Direct methods
  - ICET (Turney, 1995)
  - Cost-sensitive decision trees (Drummond & Holte, 2003; Ling et al, 2004, 2006b)
- Meta-learning
  - Thresholding
    - MetaCost (Domingos, 1999)
    - CostSensitiveClassifier *(CSC in short)* (Witten & Frank, 2005)

- Cost-sensitive naïve Bayes (Chai et al., 2004)
- Empirical Threshold Adjusting (*ETA* in short) (Sheng & Ling, 2006)

o Sampling
- Costing (Zadronzny et al., 2003)
- Weighting (Ting, 1998)

MAIN FOCUS

In this section, we will first discuss the general theory of cost-sensitive learning. Then, we will provide an overview of the works on cost-sensitive learning, focusing on cost-sensitive meta-learning.

Theory of Cost-Sensitive Learning

In this section, we summarize the theory of cost-sensitive learning, published mostly in (Elkan, 2001; Zadrozny & Elkan, 2001). The theory describes how the misclassification cost plays its essential role in various cost-sensitive learning algorithms.

Without loss of generality, we assume binary classification (i.e., positive and negative class) in this paper. In cost-sensitive learning, the costs of false positive (actual negative but predicted as positive; denoted as (*FP*), false negative (*FN*), true positive (*TP*) and true negative (*TN*) can be given in a cost matrix, as shown in Table 1. In the table, we also use the notation *C(i, j)* to represent the misclassification cost of classifying an instance from its actual class *j* into the predicted class *i*. (We use 1 for positive, and 0 for negative). These misclassification cost values can be given by domain experts. In cost-sensitive learning, it is usually assumed that such a cost matrix is given and known. For multiple classes, the cost matrix can be easily extended by adding more rows and more columns.

Note that *C(i, i)* (*TP* and *TN*) is usually regarded as the "benefit" (i.e., negated cost) when an instance is predicted correctly. In addition, cost-sensitive learning is often used to deal with datasets with very imbalanced class distribution (Japkowicz, 2000; Chawla et al., 2004). Usually (and without loss of generality), the minority or rare class is regarded as the positive class, and it is often more expensive to misclassify an actual positive example into negative, than an actual negative example into positive. That is, the value of *FN* or *C(0,1)* is usually larger than that of *FP* or *C(1,0)*. This is true for the cancer example mentioned earlier (cancer patients are usually rare in the population, but predicting an actual cancer patient as negative is usually very costly) and the bomb example (terrorists are rare).

Given the cost matrix, an example should be classified into the class that has the minimum expected cost. This is the minimum expected cost principle (Michie, Spiegelhalter, & Taylor, 1994). The expected cost *R(i|x)* of classifying an instance *x* into class *i* (by a classifier) can be expressed as:

$$R(i \mid x) = \sum_j P(j \mid x) C(i, j), \quad (1)$$

where *P(j|x)* is the probability estimation of classifying an instance into class *j*. That is, the classifier will classify an instance *x* into positive class if and only if:

$$P(0|x)C(1,0) + P(1|x)C(1,1) \leq P(0|x)C(0,0) + P(1|x)C(0,1)$$

This is equivalent to:

$$P(0|x)(C(1,0)-C(0,0)) \leq P(1|x)(C(0,1)-C(1,1))$$

Table 1. An example of cost matrix for binary classification

	Actual negative	Actual positive
Predict negative	C(0,0), or TN	C(0,1), or FN
Predict positive	C(1,0), or FP	C(1,1), or TP

Table 2. A simpler cost matrix with an equivalent optimal classification

	True negative	True positive
Predict negative	0	*C(0,1) - C(1,1)*
Predict positive	*C(1,0) - C(0,0)*	0

Thus, the decision (of classifying an example into positive) will not be changed if a constant is added into a column of the original cost matrix. Thus, the original cost matrix can always be converted to a simpler one by subtracting *C(0,0)* from the first column, and *C(1,1)* from the second column. After such conversion, the simpler cost matrix is shown in Table 2. Thus, any given cost-matrix can be converted to one with *C(0,0) = C(1,1) = 0*. This property is stronger than the one: the decision is not changed if a constant is added to each entry in the matrix (Elkan, 2001). In the rest of the paper, we will assume that *C(0,0) = C(1,1) = 0*. Under this assumption, the classifier will classify an instance *x* into positive class if and only if:

$$P(0|x)C(1,0) \leq P(1|x)C(0,1)$$

As $P(0|x)=1- P(1|x)$, we can obtain a threshold p^* for the classifier to classify an instance *x* into positive if $P(1|x) \geq p^*$, where

$$p^* = \frac{C(1,0)}{C(1,0)+C(0,1)}. \quad (2)$$

Thus, if a cost-insensitive classifier can produce a posterior probability estimation $p(1|x)$ for test examples *x*, we can make it cost-sensitive by simply choosing the classification threshold according to (2), and classify any example to be positive whenever $P(1|x) \geq p^*$. This is what several cost-sensitive meta-learning algorithms, such as Relabeling, are based on (see later for details). However, some cost-insensitive classifiers, such as C4.5 (Quinlan, 1993), may not be able to produce accurate probability estimation; they are designed to predict the class correctly. *ETA* (Sheng & Ling, 2006) does not require accurate estimation of probabilities – an accurate ranking is sufficient. *ETA* simply uses cross-validation to search the best probability from the training instances as the threshold.

Traditional cost-insensitive classifiers are designed to predict the class in terms of a default, fixed threshold of 0.5. (Elkan, 2001) shows that we can "rebalance" the original training examples by sampling such that the classifiers with the 0.5 threshold is equivalent to the classifiers with the p* threshold as in (2), in order to achieve cost-sensitivity. The rebalance is done as follows. If we keep all positive examples (as they are assumed as the rare class), then the number of negative examples should be multiplied by $C(1,0)/C(0,1) = FP/FN$. Note that as usually $FP < FN$, the multiple is less than 1. This is thus often called "under-sampling the majority class". This is also equivalent to "proportional sampling", where positive and negative examples are sampled by the ratio of:

$$p(1)\ FN : p(0)\ FP \quad (3)$$

where $p(1)$ and $p(0)$ are the prior probability of the positive and negative examples in the original training set. That is, the prior probabilities and the costs are interchangeable: doubling $p(1)$ has the same effect as doubling *FN*, or halving *FP* (Drummond & Holte, 2000). Most sampling meta-learning methods, such as Costing (Zadronzny et al., 2003), are based on (3) above (see later for details).

Almost all meta-learning approaches are either based on (2) or (3) for the thresholding- and sampling-based meta-learning methods respectively.

Cost-Sensitive Learning Algorithms

As mentioned in the Introduction, many cost-sensitive learning can be categorized into two categories. One is direct cost-sensitive learning, and the other is cost-

sensitive meta-learning. In this section, we will review typical cost-sensitive learning algorithms, focusing on cost-sensitive meta-learning.

Direct Cost-sensitive Learning. The main idea of building a direct cost-sensitive learning algorithm is to directly introduce and utilize misclassification costs into the learning algorithms. There are several works on direct cost-sensitive learning algorithms, such as ICET (Turney, 1995) and cost-sensitive decision trees (Ling et al., 2006b).

ICET (Turney, 1995) incorporates misclassification costs in the fitness function of genetic algorithms. On the other hand, cost-sensitive decision tree (Ling et al., 2006b), called *CSTree* here, uses the misclassification costs directly in its tree building process. That is, instead of minimizing entropy in attribute selection as in C4.5 (Quinlan, 1993), *CSTree* selects the best attribute by the expected total cost reduction. That is, an attribute is selected as a root of the (sub)tree if it minimizes the total misclassification cost.

Note that both ICET and *CSTree* directly take costs into model building. Besides misclassification costs, they can also take easily attribute costs (and perhaps other costs) directly into consideration, while cost-sensitive meta-learning algorithms generally cannot.

(Drummond & Holte, 2000) investigates the cost-sensitivity of the four commonly used attribute selection criteria of decision tree learning: accuracy, Gini, entropy, and DKM (Kearns & Mansour, 1996; Dietterich et al., 1996). They claim that the sensitivity of cost is highest with the accuracy, followed by Gini, entropy, and DKM.

Cost-Sensitive Meta-Learning. Cost-sensitive meta-learning converts existing cost-insensitive classifiers into cost-sensitive ones without modifying them. Thus, it can be regarded as a middleware component that pre-processes the training data, or post-processes the output, from the cost-insensitive learning algorithms.

Cost-sensitive meta-learning can be further classified into two main categories: thresholding and sampling, based on (2) and (3) respectively, as discussed in the theory of cost-sensitive learning (Section 2).

Thresholding uses (2) as a threshold to classify examples into positive or negative if the cost-insensitive classifiers can produce probability estimations. MetaCost (Domingos, 1999) is a thresholding method. It first uses bagging on decision trees to obtain accurate probability estimations of training examples, relabels the classes of training examples according to (2), and then uses the relabeled training instances to build a cost-insensitive classifier. *CSC* (Witten & Frank, 2005) also uses (2) to predict the class of test instances. More specifically, *CSC* uses a cost-insensitive algorithm to obtain the probability estimations $P(j|x)$ of each test instance. Then it uses (2) to predict the class label of the test examples. Cost-sensitive naïve Bayes (Chai et al., 2004) uses (2) to classify test examples based on the posterior probability produced by the naïve Bayes.

As we have seen, all thresholding-based meta-learning methods rely on accurate probability estimations of $p(1|x)$ for the test example x. To achieve this, (Zadrozny & Elkan, 2001) propose several methods to improve the calibration of probability estimates. Still, as the true probabilities of the training examples are usually not given, accurate estimation of such probabilities remains elusive.

ETA (Sheng & Ling, 2006) is an empirical thresholding-based meta-learning method. It does not require accurate estimation of probabilities. It works well on the base cost-insensitive classification algorithms that do not generate accurate estimation of probabilities. The only constrain is that the base cost-insensitive algorithm can generate an accurate ranking, like C4.5. The main idea of *ETA* is simple. It uses cross-validation to search the best probability from the training instances as the threshold, and uses the searched threshold to predict the class label of test instances. To reduce overfitting, *ETA* searches for the best probability as threshold from the validation sets. More specifically, an *m-fold* cross-validation is applied on the training set, and the classifier predicts the probability estimates on the validation sets.

On the other hand, sampling first modifies the class distribution of training data according to (3), and then applies cost-insensitive classifiers on the sampled data directly. There is no need for the classifiers to produce probability estimations, as long as it can classify positive or negative examples accurately. (Zadronzny et al., 2003) show that proportional sampling with replacement produces duplicated cases in the training, which in turn produces overfitting in model building. However, it is unclear if proper overfitting avoidance (without overlapping between the training and pruning sets) would work well (future work). Instead, (Zadronzny et al., 2003) proposes to use "rejection sampling" to avoid duplication. More specifically, each instance in

the original training set is drawn once, and accepted into the sample with the accepting probability $C(j,i)/Z$, where $C(j,i)$ is the misclassification cost of class i, and Z is an arbitrary constant such that $Z \geq maxC(j,i)$. When $Z = maxC(j,i)$, this is equivalent to keeping all examples of the rare class, and sampling the majority class without replacement according to $C(1,0)/C(0,1)$ – in accordance with (3). With a larger Z, the sample S' produced by rejection sampling can become much smaller than the original training set S (i.e. $|S'| << |S|$). Thus, the learning models built on the reduced sample S' can be unstable. To reduce instability, (Zadrozny et al., 2003) apply bagging (Brieman, 1996; Buhlmann & Yu, 2003; Bauer & Kohavi, 1999) after rejection sampling. The resulting method is called Costing.

Weighting (Ting, 1998) can also be viewed as a sampling method. It assigns a normalized weight to each instance according to the misclassification costs specified in (3). That is, examples of the rare class (which carries a higher misclassification cost) are assigned proportionally high weights. Examples with high weights can be viewed as example duplication – thus sampling. Weighting then induces cost-sensitivity by integrating the instances' weights directly into C4.5, as C4.5 can take example weights directly in the entropy calculation. It works whenever the original cost-insensitive classifiers can accept example weights directly. Thus, we can say that Weighting is a semi meta-learning method. In addition, Weighting does not rely on bagging as Costing does, as it "utilizes" all examples in the training set.

FUTURE TRENDS

These meta-learning methods can reuse not only all the existing cost-insensitive learning algorithms which can produce the probabilities for future predictions, but also all the improvement approaches, such as Bagging (Brieman, 1996; Bauer & Kohavi, 1999), Boosting (Schapire, 1999; Bauer & Kohavi, 1999), and Stacking (Witten & Frank, 2005). Thus, cost-sensitive meta-learning is the possible main trend of future research in cost sensitive learning. However, current cost-sensitive meta-learning algorithms only focus on minimizing the misclassification costs. In real-world applications, there are different types of costs (Turney, 2000), such as attribute costs, example costs, and misclassification costs. It is very interesting to integrate all costs together. Thus, we have to design novel cost-sensitive meta-learning algorithms with a single objection – minimizing the total cost,

CONCLUSION

Cost-sensitive learning is one of the most important topics in data mining and machine learning. Many real-world applications can be converted to cost-sensitive learning, for example, the software defect escalation prediction system (Ling et al. 2006a). All cost-sensitive learning algorithms can be categorized into directed cost-sensitive learning and cost-sensitive meta-learning. The former approach produces the specific cost-sensitive learning algorithms (such as ICET and *CSTree*). The later approach produces wrapper methods to convert cost-insensitive algorithms into cost-sensitive ones, such as Costing, MetaCost, Weighting, *CSC*, and *ETA*.

REFERENCES

Bauer, E., & Kohavi, R. (1999). An empirical comparison of voting classification algorithms: bagging, boosting and variants. *Machine Learning*, 36(1/2), 105-139.

Brieman, L. (1996). Bagging predictors. *Machine Learning*, 24, 123-140.

Buhlmann, P., & Yu, B. (2003). Analyzing bagging. *Annals of Statistics*.

Chai, X., Deng, L., Yang, Q., & Ling,C.X.. (2004). Test-Cost Sensitive Naïve Bayesian Classification. *Proceedings of the Fourth IEEE International Conference on Data Mining*. Brighton, UK : IEEE Computer Society Press.

Chawla, N.V., Japkowicz, N., & Kolcz, A. eds. (2004). *Special Issue on Learning from Imbalanced Datasets. SIGKDD, 6(1)*, ACM Press.

Dietterich, T.G., Kearns, M., & Mansour, Y. (1996). Applying the weak learning framework to understand and improve C4.5. *Proceedings of the Thirteenth International Conference on Machine Learning,* 96-104. San Francisco: Morgan Kaufmann.

Domingos, P. (1999). MetaCost: A general method for making classifiers cost-sensitive. *In Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining*, 155-164, ACM Press.

Drummond, C., & Holte, R. (2000). Exploiting the cost (in)sensitivity of decision tree splitting criteria. *Proceedings of the 17th International Conference on Machine Learning*, 239-246.

Drummond, C., & Holte, R.C. (2003). C4.5, Class Imbalance, and Cost Sensitivity: Why under-sampling beats over-sampling. *Workshop on Learning from Imbalanced Datasets II*.

Elkan, C. (2001). The Foundations of Cost-Sensitive Learning. *Proceedings of the Seventeenth International Joint Conference of Artificial Intelligence*, 973-978. Seattle, Washington: Morgan Kaufmann.

Japkowicz, N. (2000). The Class Imbalance Problem: Significance and Strategies. *Proceedings of the 2000 International Conference on Artificial Intelligence, 111-117*.

Kearns, M., & Mansour, Y. (1996). On the boosting ability of top-down decision tree learning algorithms. *Proceedings of the Twenty-Enighth ACM Symposium on the Theory of Computing,* 459-468. New York: ACM Press.

Ling, C.X., Yang, Q., Wang, J., & Zhang, S. (2004). Decision Trees with Minimal Costs. *Proceedings of 2004 International Conference on Machine Learning (ICML'2004)*.

Ling, C.X., Sheng, V.S., Bruckhaus T., & Madhavji, N.H. (2006a) Maximum Profit Mining and Its Application in Software Development. *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD06), 929-934*. August 20-23, Philadelphia, USA.

Ling, C.X., Sheng, V.S., & Yang, Q. (2006b). Test Strategies for Cost-Sensitive Decision Trees. *IEEE Transactions of Knowledge and Data Engineering, 18(8), 1055-1067*.

Lizotte, D., Madani, O., & Greiner R. (2003). Budgeted Learning of Naïve-Bayes Classifiers. *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence*. Acapulco, Mexico: Morgan Kaufmann.

Michie, D., Spiegelhalter, D.J., & Taylor, C.C. (1994). *Machine Learning, Neural and Statistical Classification*. Ellis Horwood Limited.

Quinlan, J.R. eds. (1993). *C4.5: Programs for Machine Learning.* Morgan Kaufmann.

Schapire, R.E. (1999). A brief introduction to boosting. *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*.

Sheng, V.S., & Ling, C.X. (2006). Thresholding for Making Classifiers Cost-sensitive. *Proceedings of the 21st National Conference on Artificial Intelligence, 476-481.* July 16–20, 2006, Boston, Massachusetts.

Ting, K.M. (1998). Inducing Cost-Sensitive Trees via Instance Weighting. *Proceedings of the Second European Symposium on Principles of Data Mining and Knowledge Discovery*, 23-26. Springer-Verlag.

Turney, P.D. (1995). Cost-Sensitive Classification: Empirical Evaluation of a Hybrid Genetic Decision Tree Induction Algorithm. *Journal of Artificial Intelligence Research*, 2, 369-409.

Turney, P.D. (2000). Types of cost in inductive concept learning. *Proceedings of the Workshop on Cost-Sensitive Learning at the Seventeenth International Conference on Machine Learning*, Stanford University, California.

Witten, I.H., & Frank, E. (2005). *Data Mining – Practical Machine Learning Tools and Techniques with Java Implementations.* Morgan Kaufmann Publishers.

Zadrozny, B., & Elkan, C. (2001). Learning and Making Decisions When Costs and Probabilities are Both Unknown. *Proceedings of the Seventh International Conference on Knowledge Discovery and Data Mining*, 204-213.

Zadrozny, B., Langford, J., & Abe, N. (2003). Cost-sensitive learning by Cost-Proportionate instance Weighting. *Proceedings of the 3th International Conference on Data Mining*.

KEY TERMS

Binary Classification: A typical classification task of data mining that classifies examples or objects into

two known classes, in terms of the attribute values of each example or object.

Classification: One of the task of data mining that categories examples or objects into a set of known classes, in terms of the attribute values of each example or object.

Cost-Insensitive Learning: A type of learning in data mining that does not take the misclassification costs into consideration. The goal of this type of learning is to pursue a high accuracy of classifying examples into a set of known classes.

Cost Matrix: A type of matrix that each cell in the matrix represents the cost of misclassifying an example from its true class (shown in column) into a predicted class (shown in row).

Cost-Sensitive Learning: A type of learning in data mining that takes the misclassification costs (and possibly other types of cost) into consideration. The goal of this type of learning is to minimize the total cost. The key difference between cost-sensitive learning and cost-insensitive learning is that cost-sensitive learning treats the different misclassifications differently.

Cost-Sensitive Meta-Learning: A type of cost-sensitive learning that does not change the base cost-insensitive learning algorithms, but works as a wrapper on these cost-insensitive learning algorithms, to make them cost-sensitive.

Misclassification Cost: The cost of classifying an example into an incorrect class.

Threshold: The probability value in classification that serves as the criterion for assigning an example to one of a set of known classes.

Count Models for Software Quality Estimation

Kehan Gao
Eastern Connecticut State University, USA

Taghi M. Khoshgoftaar
Florida Atlantic University, USA

INTRODUCTION

Timely and accurate prediction of the quality of software modules in the early stages of the software development life cycle is very important in the field of software reliability engineering. With such predictions, a software quality assurance team can assign the limited quality improvement resources to the needed areas and prevent problems from occurring during system operation. Software metrics-based quality estimation models are tools that can achieve such predictions. They are generally of two types: a classification model that predicts the class membership of modules into two or more quality-based classes (Khoshgoftaar et al., 2005b), and a quantitative prediction model that estimates the number of faults (or some other quality factor) that are likely to occur in software modules (Ohlsson et al., 1998).

In recent years, a variety of techniques have been developed for software quality estimation (Briand et al., 2002; Khoshgoftaar et al., 2002; Ohlsson et al., 1998; Ping et al., 2002), most of which are suited for either prediction or classification, but not for both. For example, logistic regression (Khoshgoftaar & Allen, 1999) can only be used for classification, whereas multiple linear regression (Ohlsson et al., 1998) can only be used for prediction. Some software quality estimation techniques, such as case-based reasoning (Khoshgoftaar & Seliya, 2003), can be used to calibrate both prediction and classification models, however, they require distinct modeling approaches for both types of models. In contrast to such software quality estimation methods, count models such as the Poisson regression model (PRM) and the zero-inflated Poisson (ZIP) regression model (Khoshgoftaar et al., 2001) can be applied to yield both with just one modeling approach. Moreover, count models are capable of providing the probability that a module has a given number of faults. Despite the attractiveness of calibrating software quality estimation models with count modeling techniques, we feel that their application in software reliability engineering has been very limited (Khoshgoftaar et al., 2001). This study can be used as a basis for assessing the usefulness of count models for predicting the number of faults and quality-based class of software modules.

BACKGROUND

Software Metrics and Software Quality Modeling

Software product and process metrics are essential in the software development process. With metrics, the software development team is able to evaluate, understand, monitor and control a software product or its development process from original specifications all the way up to implementation and customer usage.

In the software reliability engineering literature, the relationship between software complexity metrics and the occurrence of faults in program modules has been used by various metrics-based software quality estimation models, such as case-based reasoning (Khoshgoftaar & Seliya, 2003), regression trees (Khoshgoftaar et al., 2002), fuzzy logic (Xu et al., 2000), genetic programming (Liu & Khoshgoftaar, 2001) and multiple linear regression (Ohlsson et al., 1998). Typically, a software quality model for a given software system is calibrated using the software metrics and fault data collected from a previous system release or similar project. The trained model can then be applied to predict the software quality of a currently under-development release or comparable project. Subsequently, the resources allocated for software quality improvement initiatives can then be targeted toward program modules that are of low quality or are likely to have many faults.

Count Modeling Techniques

Count models have many applications in economics, medical and health care, and social science (Cameron & Windmeijer, 1996; Deb & Trivedi, 1997; Mullahy, 1997). Count models refer to those models where the values of the dependent variable are count numbers, i.e., zeros or positive integers. The typical count models include Poisson regression models (Khoshgoftaar et al., 2005a), negative binomial regression models (Cameron & Trivedi, 1998), hurdle models (Gurmu, 1997) and zero-inflated models (Lambert, 1992).

The Poisson regression method is the basis in count modeling approach. It requires *equidispersion*, i.e., equality of mean and variance of the dependent variable. However in real life systems, the variance of the dependent variable often exceeds its mean value. This phenomenon is known as *overdispersion*. In software quality estimation models, overdispersion is often displayed by an excess of zeros for the dependent variable, such as number of faults. In such cases, a pure PRM fails to make an accurate quality prediction. In order to overcome this limitation of the pure PRM, a few but limited numbers of modifications or alternatives to the pure PRM have been investigated.

Mullahy (1986) used a with-zeros (WZ) model instead of a standard PRM. In his study, it was assumed that the excess of zeros resulted from a mixture of two processes, both of which produced zeros. One process generated a binary outcome, for example: perfect or non-perfect, while the other process generated count quantities which might follow some standard distribution such as, Poisson or negative binomial. Consequently, the mean structure of the pure Poisson process was changed to the weighted combination of the means from each process.

Lambert (1992) used a similar idea to Mullahy's when introducing the zero-inflated Poisson (ZIP) model. An improvement of ZIP over WZ is that ZIP establishes a logit relationship between the "probability that a module is perfect" and the "independent variables of the data set". This relationship ensures that the probability distribution is between 0 and 1.

Similar to the zero-inflated model, another variation of count models is the hurdle model (Gurmu, 1997) which also consists of two parts. However, instead of having perfect and non-perfect groups of modules, the hurdle model divides them into a lower count group and a higher count group, based on a binary distribution. The dependent variable of each of these groups is assumed to follow a separate distribution process. Therefore, two factors may affect predictive quality of the hurdle models: the crossing or threshold value of the dependent variable that is used to form the two groups, and the specific distribution each group is assumed to follow.

Preliminary studies (Khoshgoftaar et al., 2001; Khoshgoftaar et al., 2005a) performed by our research team examined the ZIP regression model for software quality estimation. In the studies, we investigated software metrics and fault data collected for all the program modules from a commercial software system. It was found that over two thirds of the program modules had no faults. Therefore, we considered the ZIP regression method to develop the software quality estimation model.

MAIN FOCUS

Poisson Regression Model

The Poisson regression model is derived from the Poisson distribution by allowing the expected value of the dependent variable to be a function associated with the independent variables. Let $(y_i, \mathbf{x}_i)$ be an observation in a data set, such that y_i and $\mathbf{x}_i$ are the dependent variable and vector of independent variables for the i^{th} observation. Given $\mathbf{x}_i$, assume y_i is Poisson distributed with the probability mass function (*pmf*) of,

$$\Pr(y_i \mid \mu_i, \mathbf{x}_i) = \frac{e^{-\mu_i}\mu_i^{y_i}}{y_i!} \quad \text{for } y_i = 0, 1, 2, \cdots, \tag{1}$$

where μ_i is the mean value of the dependent variable y_i. The expected value (E) and the variance (Var) of y_i are identical, i.e., $E(y_i|\mathbf{x}_i) = Var(y_i|\mathbf{x}_i) = \mu_i$.

To ensure that the expected value of y_i is nonnegative, the link function, which displays a relationship between the expected value and the independent variables, should have the following form (Cameron & Trivedi, 1998):

$$\mu_i = E(y_i|\mathbf{x}_i) = e^{\mathbf{x}_i'\boldsymbol{\beta}} \tag{2}$$

where $\boldsymbol{\beta}$ is an unknown parameter vector and $\mathbf{x}_i'$ represents the transpose of the vector $\mathbf{x}_i$.

Zero-Inflated Poisson Regression Model

The zero-inflated Poisson regression model assumes that all zeros come from two sources: the source representing the perfect modules in which no faults occur, and the source representing the non-perfect modules in which the number of faults in the modules follows the Poisson distribution. In the ZIP model, a parameter, φ_i, is introduced to denote the probability of a module being perfect. Hence, the probability of the module being non-perfect is $1 - \varphi_i$. The *pmf* of the ZIP model is given by,

$$\Pr(y_i \mid \mathbf{x}_i, \mu_i \phi_i) = \begin{cases} \phi_i + (1-\phi_i)e^{-\mu_i}, & y_i = 0, \\ (1-\phi_i)\dfrac{e^{-\mu_i}\mu_i^{y_i}}{y_i!}, & y_i = 1, 2, 3, \cdots. \end{cases} \quad (3)$$

The ZIP model is obtained by adding the next two link functions:

$$\log(\mu_i) = \mathbf{x}_i' \boldsymbol{\beta} \quad (4)$$

$$\text{logit}(\phi_i) = \log\frac{\phi_i}{1-\phi_i} = \mathbf{x}_i' \boldsymbol{\gamma}, \quad (5)$$

where log denotes the natural logarithm and $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ are unknown parameter vectors.

The maximum likelihood estimation (MLE) technique is used in the parameter estimation for the PRM and ZIP models (Khoshgoftaar et al., 2005a).

Software Quality Classification with Count Models

Software quality classification models are initially trained with known software metrics (independent variables) and class membership (dependent variable) data for the system being modeled. The dependent variable is indicated by class, i.e., fault-prone (*fp*) and not fault-prone (*nfp*). The quality-based class membership of the modules in the training data set is usually assigned based on a pre-determined threshold value of a quality factor, such as number of faults or number of lines of code churn. For example, a module is defined as *fp* if its number of faults exceeds the selected threshold value, and *nfp* otherwise. The chosen threshold value is usually dependent on the project management's quality improvement objectives. In a two-group classification model, two types of misclassifications can occur: Type I (*nfp* module classified as *fp*) and Type II (*fp* module classified as *nfp*).

We present a generic approach to building count models for classification. Given a data set with observations, $(y_i, \mathbf{x}_i)$, the following steps illustrate the approach for classifying software modules as *fp* and *nfp*:

1. Estimate probabilities of the dependent variable (number of faults) being distinct counts by the given *pmf*, $\hat{\Pr}(y_i)$, $y_i = 0, 1, 2, \ldots$.
2. Calculate the probabilities of a module being *fp* and *nfp*. The probability of a module being *nfp*, i.e., the probability of a module's dependent variable being less than the selected threshold (t_0), is given by $f_{nfp}(\mathbf{x}_i) = \sum_{y_i < t_0} \hat{\Pr}(y_i)$. Then, the probability of a module being *fp* is $f_{fp}(\mathbf{x}_i) = 1 - \sum_{y_i < t_0} \hat{\Pr}(y_i)$.
3. Apply the generalized classification rule (Khoshgoftaar & Allen, 2000) to obtain the class membership of all modules.

$$Class(\mathbf{x}_i) = \begin{cases} fault-prone, & \text{if } \frac{f_{fp}(\mathbf{x}_i)}{f_{nfp}(\mathbf{x}_i)} \geq c \\ not\ fault-prone, & \text{otherwise.} \end{cases} \quad (6)$$

where c is a model parameter, which can be varied to obtain a model with the preferred balance between the Type I and Type II error rates. The preferred balance is dependent on the quality improvement needs of the given project.

Quantitative Software Quality Prediction with Count Models

For count models, the expected value of the dependent variable is used to predict the number of faults for each module. The *pmf* is used to compute the probability of each module having various counts of faults (known as *prediction probability*). Prediction probability can be useful in assessing the uncertainty and risks associated with software quality prediction models.

The accuracy of fault prediction models are measured by the average absolute error (AAE) and the av-

erage relative error (ARE), i.e., $\text{AAE} = \frac{1}{n}\sum_{i=1}^{n} | \hat{y}_i - y_i |$; $\text{ARE} = \frac{1}{n}\sum_{i=1}^{n} \frac{|\hat{y}_i - y_i|}{y_i + 1}$, where n is the number of modules, y_i and $\hat{y}_i$ are the actual and predicted values of the dependent variable. In the case of ARE, since the actual value of the dependent variable may be zero, we add a '1' to the denominator to avoid division by zero.

A CASE STUDY

System Description

The software metrics and fault data for this case study were collected from a Command and Control Communication System (*CCCS*), which is a large military telecommunications system written in Ada using the procedural paradigm. Software product metrics were collected from the source code. A software module was identified as an Ada package, consisting of one or more procedures. The software product metrics used to model the software quality of this system are presented in Table 1. The dependent variable, *Fault*, was attributed to each module during the system integration and test phase. The top 20% of the modules contained 82.2% of the faults and 52% of the modules had no faults. The data set consisted of 282 program modules. A software module was considered as *fp* if it had at least four faults and *nfp* otherwise.

Calibrating Models

Using an impartial data splitting technique, we designated the 282 modules into the Fit (188) and Test (94) data sets for model-fitting and model-evaluation. The performance evaluation of software quality models on the test data set simulates the predictive capability of the models when applied to an under-development system release or similar project.

Principle components analysis (PCA) (Berenson et al., 1983) was used to remove existing correlation among the software product metrics and also to reduce the dimensionality of the multivariate data set. Two significant principle components were extracted from the original eight software metrics of the *CCCS* system. These two principle components accounted for about 94% of the explained variance in the data set and accordingly they are used to calibrate classification and prediction models for this case study. The estimated parameters based on the two principle components for PRM and ZIP are summarized in Table 2.

Results

Our modeling objective for the classification models was to obtain a model that depicted the preferred balance of equality between the error rates, with Type II being as low as possible. This model selection strategy was based on our discussions with the software development team of the telecommunications system. We have summarized the predictive performances (based on the Test data set) of the preferred classification models of PRM and ZIP in Table 3. We observe that though the predictive accuracies (numbers of Type I and Type II errors) of the two models are similar, the ZIP model is the closest match to our model selection strategy. Moreover, the overall misclassification error rate of the ZIP model is also slightly lower than that of PRM.

For software fault prediction, Table 4 presents both the quality-of-fit and the predictive abilities of the two count models. The table also shows standard deviations of the absolute error (std AE) and the relative error (std RE). A relative comparison suggests that the ZIP model has a better quality-of-fit and predictive power than the PRM. A further in-depth discussion on the relative

Table 1. Software product metrics for CCCS

Symbol	Description
η1	Number of unique operators
η2	Number of unique operands
N1	Total number of operators
N2	Total number of operands
V (G)	McCabe's cyclomatic complexity
NL	Number of logical operators
LOC	Lines of code
ELOC	Executable Lines of code

Table 2. Regression count models for CCCS

Models	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\gamma}_0$	$\hat{\gamma}_1$	$\hat{\gamma}_2$
PRM	0.2739	0.8971	-0.1398	-	-	-
ZIP	0.8714	0.6624	-0.0834	-0.6473	-2.0594	-1.2135

Table 3. Misclassification rates for CCCSTest

Model	Type I errors		Type II errors		Overall	
Type	Number	%	Number	%	Number	%
PRM	13	17.33%	3	15.79%	16	17.02%
ZIP	12	16.00%	3	15.79%	15	15.96%

performance of these two count models is presented in our recent work (Khoshgoftaar et al., 2005a). The prediction probabilities of each module having various counts of faults were computed by substituting the respective parameters of Table 2 in the *pmf* of the respective count models.

FUTURE TRENDS

Future work related to software quality modeling with count models will involve additional empirical case studies, and investigation of other count modeling techniques and comparing their classification and fault prediction performances with those of the PRM and ZIP models. The complexity of the count models' calibration (estimation of parameters) will greatly increase as the number of metrics involved in the modeling process increases. PCA is a statistical technique that can transform the original set of correlated variables into a smaller set of uncorrelated variables that are linear combinations of the original ones. However, PCA can not detect software metrics that significantly contribute to software quality. Therefore, the study that investigates the attribute selection problem for reducing the number of the software metrics in the context of count models is another interesting future research topic.

CONCLUSION

Software metrics-based quality estimation models include those that provide quality-based classification of program modules and those that provide quantitative prediction of a quality factor for the program modules. Generally speaking, software quality modeling techniques are suited for only one of these types of models, i.e., classification or prediction, but not for both. In contrast, count models can be adapted to calibrate both classification models and prediction models. Moreover, their unique ability to provide the probability that a given program module will have a specific number of faults can not be obtained by other prediction techniques. A promising use of such information is to assess the uncertainty and risks of software quality predictions obtained by the count models.

Since the number of faulty modules observed in high assurance systems, such as telecommunications systems, is usually a small fraction of the total number of modules, the ZIP model is suited for the software quality modeling of such systems: it uses a different process to model the program modules with no faults.

REFERENCES

Berenson, M. L., Levine, M., & Goldstein, M (1983). *Intermediate Statistical Methods and Applications: A*

Table 4. Prediction accuracy of count models for CCCS

Data	Fit				Test			
Statistics	AAE	ARE	std AE	std RE	AAE	ARE	std AE	std RE
PRM	1.7513	0.6877	2.7726	0.5098	1.9961	0.7327	2.9963	0.6214
ZIP	1.6182	0.629	2.4043	0.583	1.7501	0.6847	2.4713	0.7493

Computer Package Approach. Englewood Cliffs, NJ: Prentice Hall.

Briand, L. C., Melo, W. L., & Wust, J. (2002). Assessing the applicability of fault-proneness models across object-oriented software projects. *IEEE Transactions on Software Engineering, 28*(7):706–720.

Cameron, A. C., & Trivedi, P. K. (1998). *Regression Analysis of Count Data*. Cambridge University Press.

Cameron, A. C., & Windmeijer, F. A. G. (1996). R-squared measures for count data regression models with applications to health-care utilization. *Journal of Business and Economic Statistics, 14*(2):209–220.

Deb, P., & Trivedi, P. K. (1997). Demand for medical care by the elderly: A finite mixture approach. *Journal of Applied Econometrics, 12*:313–336.

Gurmu, S. (1997). Semi-parametric estimation of hurdle regression models with an application to medicaid utilization. *Journal of Applied Econometrics, 12*(3):225–242.

Khoshgoftaar, T. M., & Allen, E. B. (1999). Logistic regression modeling of software quality. International Journal of Reliability, Quality and Safety Engineering, 6(4):303–317.

Khoshgoftaar, T. M., & Allen, E. B. (2000). A practical classification rule for software quality models. *IEEE Transactions on Reliability, 49*(2):209–216.

Khoshgoftaar, T. M., Allen, E. B., & Deng, J. (2002). Using regression trees to classify fault-prone software modules. *IEEE Transactions on Reliability, 51*(4):455–462.

Khoshgoftaar, T. M., Gao, K., & Szabo, R. M. (2001). An application of zero-inflated Poisson regression for software fault prediction. *Proceedings of the Twelfth International Symposium on Software Reliability Engineering* (pp. 66–73). IEEE Computer Society.

Khoshgoftaar, T. M., Gao, K., & Szabo, R. M. (2005a). Comparing software fault predictions of pure and zero-inflated Poisson regression models. *International Journal of Systems Science, 36*(11):705–715, 2005.

Khoshgoftaar, T. M., & Seliya, N. (2003). Analogy-based practical classification rules for software quality estimation. *Empirical Software Engineering, 8*(4):325–350.

Khoshgoftaar, T. M., Seliya, N., & Gao, K. (2005b). Assessment of a new three-group software quality classification technique: An empirical case study. *Journal of Empirical Software Engineering, 10*(2):183–218.

Lambert, D. (1992). Zero-inflated Poisson regression, with an application to defects in manufacturing. *Technometrics, 34*(1):1–14.

Liu, Y., & Khoshgoftaar, T. M. (2001). Genetic programming model for software quality classification. *Proceedings: High Assurance Systems Engineering* (pp. 127–136). IEEE Computer Society.

Mullahy, J. (1986). Specification and testing of some modified count data models. *Journal of Econometrics, 33*:341–365.

Mullahy, J. (1997). Instrumental variable estimation of count data models: Applications to models of cigarette smoking behavior. *Review of Economics and Statistics, 79*:586–593.

Ohlsson, N., Zhao, M., & Helander, M. (1998). Application of multivariate analysis for software fault prediction. *Software Quality Journal, 7*(1):51–66.

Ping, Y., Systa, T., & Muller, H. (2002). Predicting fault-proneness using OO metrics. an industrial case study. In T. Gyimothy and F. B. Abreu (Ed.), *Proceedings: 6th European Conference on Software Maintenance and Reengineering* (pp. 99–107).

Xu, Z., Khoshgoftaar, T. M., & Allen, E. B. (2000). Application of fuzzy linear regression model for predicting program faults. In H. Pham and M.-W. Lu (Ed.), *Proceedings: Sixth ISSAT International Conference on Reliability and Quality in Design* (pp. 96–101). International Society of Science and Applied Technologies.

KEY TERMS

Overdispersion: *Overdispersion* occurs when the observed variance of the data is larger than the predicted variance. This is particularly apparent in the case of a Poisson regression model.

Poisson Regression Models: Poisson regression assumes that the data follows a Poisson distribution, which is frequently encountered when counting a number of events. Poisson regression also models the variance as a function of the mean.

Software Fault Prediction: To estimate the number of faults (or some other quality factor) that are likely to occur in software modules.

Software Metrics: A software metric is a measure of some property of a piece of software or its specifications.

Software Quality Classification: To predict the class membership of modules into two or more quality-based classes. For example, fault-prone and not fault-prone.

Prediction Probability: The probability that a given number of faults will occur in any given program module.

Zero-Inflated Poisson Regression Models: ZIP models incorporate Poisson probabilities but allow the probability of a zero to be larger than what a Poisson distribution might suggest. It is a mixture distribution where zero is set with a certain probability and larger

Data Analysis for Oil Production Prediction

Christine W. Chan
University of Regina, Canada

Hanh H. Nguygen
University of Regina, Canada

Xiongmin Li
University of Regina, Canada

INTRODUCTION

An economic evaluation of a new oil well is often required, and this evaluation depends heavily on how accurately production of the well can be estimated. Unfortunately, this kind of prediction is extremely difficult because of complex subsurface conditions of reservoirs. The industrial standard approach is to use either curve-fitting methods or complex and time-consuming reservoir simulations. In this study, we attempted to improve upon the standard techniques by using a variety of neural network and data mining approaches. The approaches differ in terms of prediction model, data division strategy, method, tool used for implementation, and the interpretability of the models. The objective is to make use of the large amount of data readily available from private companies and public sources to enhance understanding of the petroleum production prediction task. Additional objectives include optimizing timing for initiation of advanced recovery processes and identifying candidate wells for production or injection.

BACKGROUND

The production of an oil well is influenced by a variety of factors, many of which are unknown and unpredictable. Core logs, drill steam test (DST), and seismic data can provide geological information about the surrounding area; however, this information alone cannot explain all the characteristics about the entire reservoir. While core analysts and reservoir engineers can analyze and interpret geoscience data from several wells with the help of numerical reservoir simulations, the process is technically difficult, time consuming and expensive in terms of both labor and computational resources.

For a quick estimation of petroleum production only, the decline curve analysis remains the most popular methods among engineers (Baker et al., 1998; Li, Horne, 2003). However, a weakness with the decline curve analysis technique is that it is difficult to foresee which equation can adequately describe production of a reservoir. Moreover, a single curve is often inadequate for describing production data generated during the entire life of the reservoir. Therefore, fitting production data to a decline curve is a difficult process and can result in unreliable predictions (El-Banbi, Wattenbarger, 1996).

To overcome the weakness of conventional decline curve analysis techniques, we adopted data mining techniques for the task of modeling non-linear production data. We compared the use of neural networks versus curve estimation technique in the prediction of oil production (Nguyen et al, 2003) and found that with sufficient training data, the neural network approach can perform better on unknown data. In our exploration on modeling production data using artificial intelligence techniques, we introduced variations along six dimensions, which will be discussed in the next section.

MAIN FOCUS: VARIATIONS IN NEURAL NETWORK MODELING APPROACHES

Prediction Model

The first step of our research was to identify the variables involved in the petroleum production prediction task. The first modeling effort included both production time series and geoscience parameters as input variables for the model. Eight factors that influence production were identified; however, since only data for

the three parameters of permeability, porosity, and first shut-in pressure were available, the three parameters were included in the model. The production rates of the three months prior to the target prediction month were also included as input variables. The number of hidden units was determined by trial and error. After training the neural network, a sensitivity test was conducted to measure the impact of each input variable on the output. The results showed that all the geoscience variables had limited (less than 5%) influence on the production prediction.

Therefore, the second modeling effort relied on a model that consists of time series data only. The training and testing error was only slightly different from those of the first model. Hence, we concluded that it is reasonable to omit the geoscience variables from our model. More details on the modeling efforts can be found in (Nguyen et al., 2004).

Data Manipulation Strategy

Since different ways of data preprocessing can influence model accuracy and usage, we investigated the following three approaches for processing the monthly oil production data (Nguyen and Chan, 2004 b): (1) **sequential**, when data from individual wells were arranged sequentially, (2) **averaging**, when data from individual wells were averaged over their lifetimes, and (3) **individual**, when data from each individual well were treated independently. Two types of models were used: one with past productions as input and another with time indices as input.

The results showed that with production volumes as input, the average and the individual approaches suffered from forecast inaccuracy when training data was insufficient: the resulting neural networks only performed well when the training data covered around 20 years of production. This posed a problem since many wells in reality cannot last that long, and it is likely that intervention measures such as water flooding would have been introduced to the well before that length of time has elapsed. Hence, these two approaches were not suitable for prediction at the initial stage of the well. On the other hand, the sequential approach used a large number of training samples from both early and late stages, therefore the resulting models worked better when provided with unseen data.

The neural networks with production months as input usually required less data for training. However, it was also observed that long-life wells generated smoother decline curves. We believe it is recommendable to use all three data preprocessing approaches, i.e. sequential, averaging and individual, when the time index is used. The individual approach can provide reference points for new infill wells while the other two approaches give some estimation of what would happen to a typical well in the study area.

Multiple Neural Networks

Several researchers have attempted to use multiple neural networks to improve model accuracy. Hashem et al. (1994) proposed using optimal linear combinations of a number of trained neural networks in order to integrate the knowledge required by the component networks. Cho and Kim (1995) presented a method using fuzzy integral to combine multiple neural networks for classification problems. Lee (1996) introduced a multiple neural network approach in which each network handles a different subset of the input data. In addition to a main processing neural network, Kadaba et al. (1989) used data-compressing neural networks to decrease the input and output cardinalities.

The multiple-neural-network (MNN) approach proposed in (Nguyen and Chan, 2004 a) aims to improve upon the classical recursive one-step-ahead neural network approach. The objective of the new approach is to reduce the number of recursions needed to reach the lead time. A MNN model is a group of neural networks working together to perform a task. Each neural network was developed to predict a different time period ahead, and the prediction terms increased at a binary exponential rate. A neural network that predict 2^n step ahead is called an n-ordered neural network.

The choice of binary exponential was made due to two reasons. First, big gaps between two consecutive neural networks are not desirable. Forecasting is a process of reducing the lead time to zero and smaller. The smaller the gaps are, the fewer steps the model needs to take in order to make a forecast. Secondly, binary exponential does not introduce bias on the roles of networks while a higher exponential puts a burden onto the lower-ordered neural networks.

To make a prediction, the neural network with the highest possible order is used first. For example, to predict 7 units ahead (x_{t+7}), a 2-ordered neural network is used first. It then calculates temporary variables backward using lower-ordered neural networks.

Implementation Tools

In order to implement our neural networks, we used two development tools from Gensym Corporation, USA, called NeurOn-Line (NOL) and NeurOnline Studio (NOL Studio). However, these tools cannot be extended to support the development of multiple neural network systems described in the previous section. Therefore, we implemented an in-house tool in Java™ for this purpose (Nguyen and Chan, 2004 a). The MNN system includes several classes that implement methods for training and testing neural networks, making forecasts, and connecting sub-neural networks together. The structure of the system is illustrated in Figure 1.

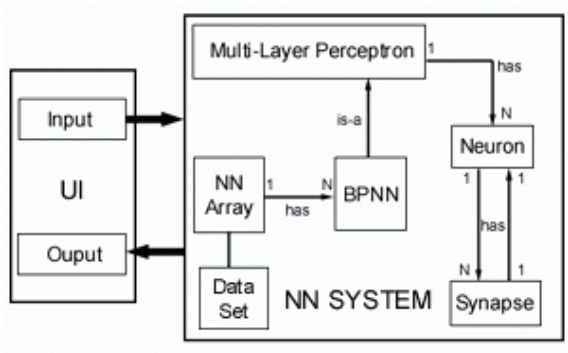
Decision Support System for Best and Worst Case Scenarios

In addition to predicting a crisp number as the production volume, we also developed a decision support system (DSS) that predicts a range of production for an in-fill well based on existing wells in the reservoir (Nguyen and Chan, 2005). The DSS consists of three main components: history-matching models, an analogue predictor, and a user interface.

The history-matching models include both neural network and curve estimation models, and they were developed offline based on historical data for each existing wells in the study area. Using the multiplicative analogue predictor, the best and worst case scenarios were calculated from these models with respect to initial condition of the new well. The system generates a table of predicted production rate and production life in months with respect to each reference well, and also graphs of the best and worse case scenarios for each method among all the wells in the group.

We carried out a test on how accurate the system can predict based on available data. Since some wells in our data set had not come to the end of their production lives, complete sets of observed data were not available. We compensated for this limitation by restricting the comparison of the predicted results of each well against those of other wells in the same group only for the available data sets. The results showed that the ranges of values that the decision support system predicts were relatively wide. A possible reason is that even though the wells come from the same area, they have different characteristics.

Figure 1. Main components of multiple neural network system



Learning Rules with Neural Based Decision Learning Model

While the artificial neural network technology is a powerful tool for predicting oil production, it cannot give explicit explanation of the result. To obtain explicit information on the processing involved in generating a prediction, we adopted the neural based decision-learning approach.

The decision-tree learning algorithms (such as C4.5) derive decision trees and rules based on the training dataset. These derived trees and rules can be used to predict the classes of new instances. Since the decision-tree learning algorithms usually work with only one attribute at a time, without considering the dependencies among attributes, the results generated by those algorithms are not optimal. However, considering multiple attributes simultaneously can introduce a combinatorial explosion problem.

In reality, the process of oil production is influenced by many interdependent factors, and an approach which can handle the complexity is needed. The Neural-Based Decision Tree Learning (NDT) proposed by (Lee and Yen 2002) combines neural network and decision tree algorithms to handle the problem. The NDT model is discussed as follows. The architecture of the NDT model for a mix-type data set, with squared boxes showing the main processing components of the neural network and decision tree is shown below. The NDT algorithm models a single pass process initialized by the acceptance of the training data as input to the neural network model. The generated output is then passed on for decision tree construction, and the resulting rules will be the net output of the NDT model. The arrows

show the direction of data flow from the input to the output, with the left-hand-side downward arrows indicating the processing stages of nominal attributes, and the right-hand-side arrows showing the corresponding process for numeric-type data.

As illustrated in Figure 2 below, the rule extraction steps in this NDT model include the following:

1. Divide the numerical-categorical-mixed dataset into two parts, e.g. numerical subset and nominal subset. For a pure-type data set, no division of the data set is necessary.
2. For the numerical subset, train a feed-forward back-propagation neural network and collect weights between the input layer and the first hidden layer. Then, change each attribute value according to the weights generated by the neural network model.
3. For the categorical subset, train a back-propagation neural network and classify categorical attributes according to the weights generated by the neural network model.
4. Combine the new numerical subset and new categorical subset into a new numerical-categorical-mixed dataset.
5. The new dataset is fed as input to the C4.5 system to generate the decision tree and rules.

Figure 2. Design of the NDT model

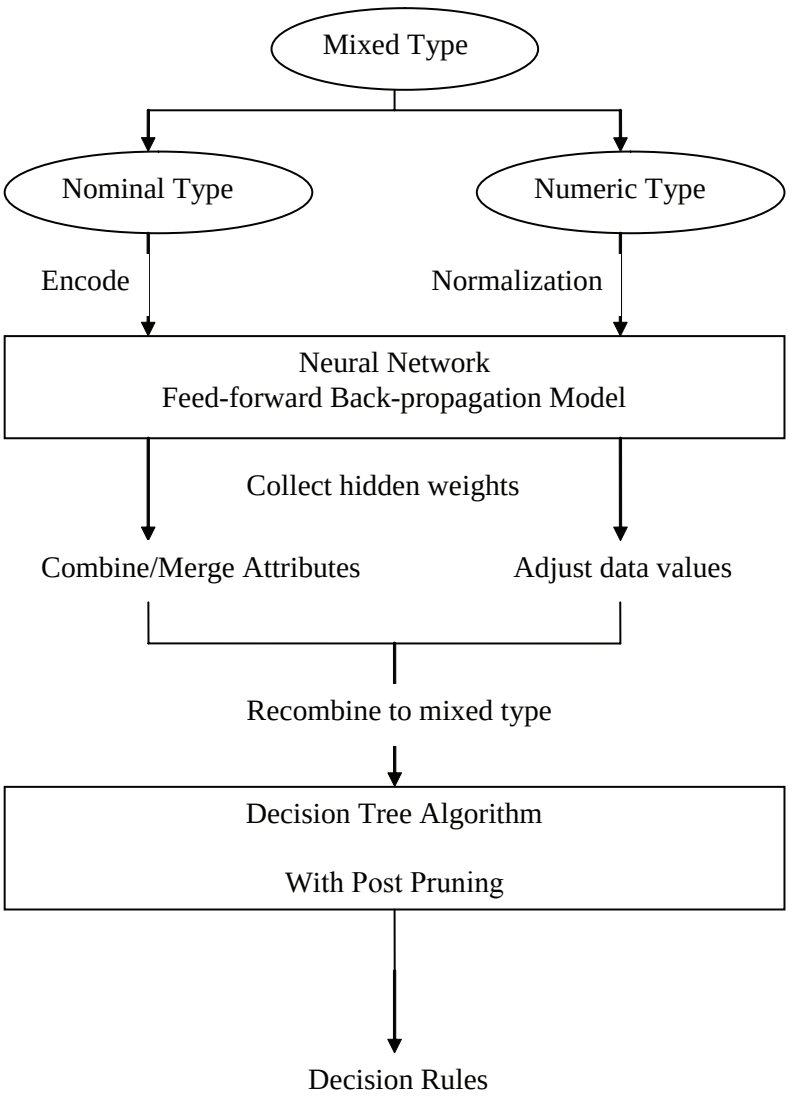
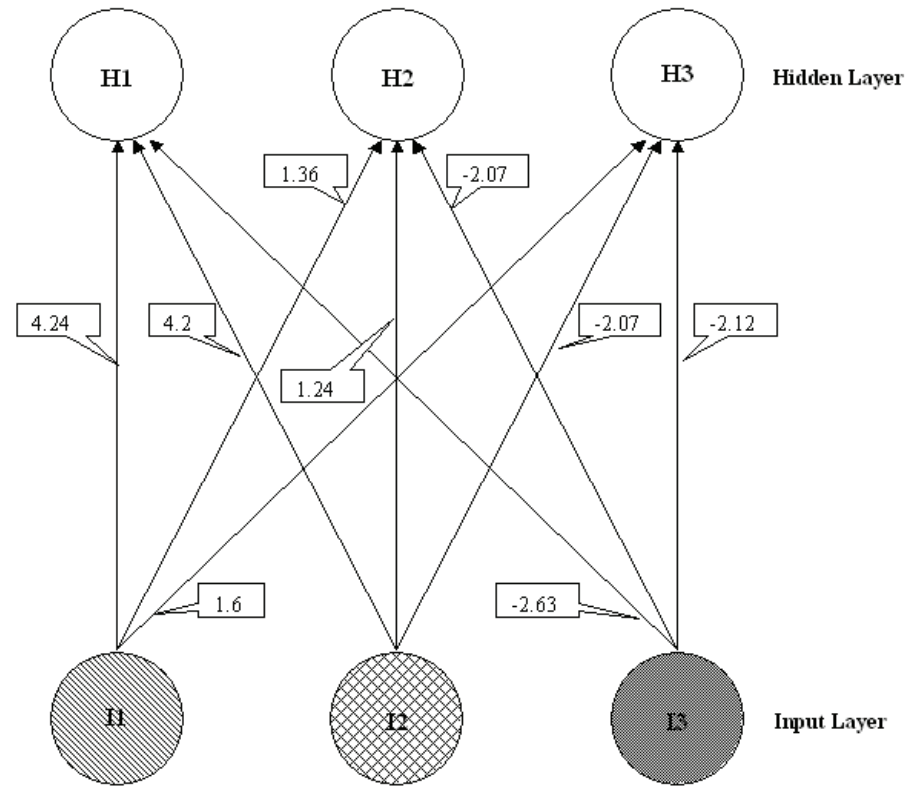


Figure 3. Training result of the original dataset



Application of NDT Model for Petroleum Production Data Classification

The proposed NDT model was developed and applied for classifying data on 320 wells from the Midale field in SaskatFchewan, Canada, Some preliminary results are presented in (Li. and Chan, 2007). A 10-fold cross-validation was used in all the decision-tree experiments, where the 10 samples were randomly selected without replacement from the training set. The dataset contains three numeric attributes that describe geoscientific properties which are significant factors in oil production: Permeability, Porosity and First-Shut-In-Pressure. With only numerical-type data in this dataset, the neural network model was trained with the following parameters:

- Number of Hidden Layers: 1
- Input Nodes: 3
- Hidden Nodes: 3
- Learning Rate: 0.3
- Momentum: 0.2
- Activation function: Sigmoid Function

The training result of the model is shown in Figure 3. Based on the result, we examine the link weights between the input layer and hidden layer. Then, the original numeric data set was trained using Eq.(1),(2), (3) shown below into a new dataset which has equivalent values as the inputs to the hidden layer of the trained neural network.

$$H_1 = -0.37 + 4.24 \times Permeability + 4.2 \times Porosity + (-2.63) \times Pressure \quad (1)$$

$$H_2 = -0.76 + 1.36 \times Permeability + 1.24 \times Porosity + (-2.07) \times Pressure \quad (2)$$

$$H_3 = -0.82 + 1.6 \times Permeability + 1.31 \times Porosity + (-2.12) \times Pressure \quad (3)$$

Figure 4. Sample rule generated by C4.5 decision-tree model

```
IF Porosity <= 11.7348 (Unit: Percent) THEN
    IF Permeability <= 4.6188 (Unit: Millidarcy) THEN
       IF Pressure <= 14893.0 (Unit: MPa) THEN
          Oil Production = Low [2000, 97767] (Unit: m³)
```

Figure 5. Sample rule generated by NDT model

```
IF H1 <= -38035.40 THEN
    IF H2 <= -30334.39 THEN
        IF H3 <= -31559.88 THEN
           Oil Production = Low [2000, 97767] (Unit: m³)
```

The decision tree model was then applied to the new dataset to generate decision rules. Some sample rules generated by the original dataset using the decision tree model are shown in Figure 4 and the rules generated with the new dataset using the NDT model are shown in Figure 5.

The results of the two applications are summarized in Table 1, where the results are the averages generated from 10-fold cross-validations for all data sets. Since the sampling used in cross-validation was randomly taken for each run, the results listed include only the ones with the best observed classification rates.

FUTURE TRENDS

Currently, we have two directions for our future work. First, we plan to work with petroleum engineers who would preprocess the geoscience data into different rock formation groups and develop one model for each group. The values for the parameters of permeability and porosity will not be averaged from all depths, instead, the average of values from one formation only will be taken. The production value will be calculated as the summation of productions from different formations of one well. The focus of the work will extend the original focus on petroleum production prediction to include the objectives of characterizing the reservoir and data workflow organization. Secondly, we plan to focus on predicting behaviors or trends of an infill well instead of directly estimating its production.

CONCLUSION

It can be seen from our works that the problem of predicting oil production presents significant challenges for artificial intelligence technologies. Although we have varied our approaches on different aspects of the solution method, such as employing different data division strategies, including different parameters in the model, and adopting different AI techniques, the results are not as accurate as desired. It seems that production depends on many factors, some of which are unknown. To further complicate the problem, some factors such as permeability, cannot be accurately measured because its value varies depending on the production location and rock formation. Our future work will involve acquiring more domain knowledge and incorporating them into our models.

ACKNOWLEDGMENT

The authors would like to acknowledge the generous support of a Strategic Grant and a Postgraduate

Table 1. Sample prediction results

Measures	C4.5 without Pruning	C4.5 with Pruning	NDT without Pruning	NDT with Pruning
Tree Size (# of nodes)	131	117	67	53
Number of Rules (# of leaves)	52	45	24	17
Test Set Size	320	320	320	320
Correct Classification Rate (%)	86.88	85.31	80.94	79.69
Misclassification Rate (%)	13.12	14.69	19.06	20.31
Mean Absolute Error	0.13	0.15	0.19	0.20
Mean Squared Error	5.51	6.90	11.63	13.20.
Computation Time (milliseconds)	15692	14070	28211	26468

Scholarship from the Natural Sciences and Engineering Research Council of Canada. We are grateful also for insightful discussions and support of Michael Monea of Petroleum Technology Research Center, and Malcolm Wilson of EnergINet at various stages of the research.

REFERENCES

Baker, R.O., Spenceley, N.K., Guo B., & Schechter D.S. (1998). *Using an analytical decline model to characterize naturally fractured reservoirs.* SPE/DOE Improve Oil Recovery Symposium, Tulsa, Oklahoma, 19-22 April. SPE 39623.

Cho, S.B., & Kim, J.H. (1995). Combining multiple neural networks by fuzzy integral for robust classification. *IEEE Transactions on Systems, Man, and Cybernetics, 25*(2), 380-384.

El-Banbi, A.H., & Wattenbarger, R.A. (1996). *Analysis of commingled tight gas reservoirs.* SPE Annual Technical Conference and Exhibition, Denver, CO, 6–9 October. SPE 36736.

Hashem, S., Schemeiser, B., & Yih, Y. (1994). Optimal Linear Combinations of Neural Networks: An Overview. *Proceedings of the 1994 IEEE International Conference on Neural Networks (ICNN'94)*, Orlando, FL.

Kadaba, N, Nygard, K.E., Juell, P.L., & Kangas, L. (1989). Modular back-propagation neural networks for large domain pattern classification. In *Proceedings of the International Joint Conference on Neural Networks IJCNN'89*, (pp. 607-610), Washington DC.

Lee, Y-S., & Yen, S-J. (2002). Neural-based approaches for improving the accuracy of decision trees. In *Proceedings of the Data Warehousing and Knowledge Discovery Conference*, 114-123.

Lee, Y-S., Yen, S-J., & Wu, Y-C. (2006). Using neural network model to discover attribute dependency for improving the performance of classification. *Journal of Informatics & Electronics, 1*(1).

Lee, B. J. (1996). Applying parallel learning models of aritificial neural networks to letters recognition from phonemes. In *Proceedings of the Conference on Integrating Multiple Learned Models for Improving and Scaling Machine Learning Algorithms* (pp. 66-71), Portland, Oregon.

Li., X, & Chan, C.W. (2007). *Towards a neural-network-based decision tree learning algorithm for petroleum production prediction*. Accepted for IEEE CCECE 2007.

Li., X, & Chan, C.W. (2006). Applying a machine learning algorithm for prediction. In *Proceedings of 2006 International Conference on Computational Intelligence and Security (CIS 2006)*, pp. 793-706.

Li., K., & Horne, R.N. (2003). *A decline curve analysis model based on fluid flow mechanisms*. SPE Western Regional/AAPG Pacific Section Joint Meeting held in Long Beach, CA. SPE 83470.

Nguyen, H.H., & Chan, C.W. (2005). Application of data analysis techniques for oil production prediction. *Engineering Applications of Artificial Intelligence. 18*(5), 549-558.

Nguyen, H.H., & Chan, C.W. (2004). Multiple neural networks for a long term time series forecast. *Neural Computing, 13*(1), 90–98.

Nguyen, H.H., & Chan, C.W. (2004b). A Comparison of Data Preprocessing Strategies for Neural Network Modeling of Oil Production Prediction. *Proceedings of The 3rd IEEE International Conference on Cognitive Informatics (ICCI)*, (pp. 199-107).

Nguyen, H.H., Chan, C.W., & Wilson, M. (2004). Prediction of oil well production: A multiple neural-network approach. *Intelligent Data Analysis, 8*(2). 183-196.

Nguyen, H.H., Chan, C.W., Wilson, M. (2003). Prediction of Petroleum Production: Artificial Neural Networks and Curve Estimation. *Proceeding of International Society for Environment Information Sciences*, Canada. 1. 375-385.

KEY TERMS

Artificial Neural Network: An interconnected group of artificial neurons (or processing units) that uses a mathematical model or computational model for information processing based on a connectionist approach to computation.

Curve Estimation: A procedure for finding a curve which matches a series of data points and possibly other constraints. It involves computing the coefficients of a function to approximate the values of the data points.

Decision Tree: A decision tree is an idea generation tool that generally refers to a graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. In data mining, a decision tree is a predictive model.

Decision Support System (DSS): A DSS is an interactive computer-based system intended to help managers make decisions. A DSS helps a manager retrieve, summarize and analyze decision relevant data.

Multiple Neural Networks: A multiple neural network system is a set of single neural networks integrated into a cohesive architecture. Multiple neural networks have been demonstrated to improve performance on tasks such as classification and regression when compared to single neural networks.

Neural-Based Decision Tree (NDT): The NDT model proposed by Lee and Yen attempts to use neural network to extract the underlying attribute dependencies that are not directly observable by humans. This model combines neural network and decision tree algorithms to handle the classification problem.

Prediction of Oil Production: A forecast on future volume of oil extracted from a production oil well based on historical production records.

Data Confidentiality and Chase-Based Knowledge Discovery

Seunghyun Im
University of Pittsburgh at Johnstown, USA

Zbigniew W. Ras
University of North Carolina, Charlotte, USA

INTRODUCTION

This article discusses data security in Knowledge Discovery Systems (KDS). In particular, we presents the problem of confidential data reconstruction by Chase (Dardzinska and Ras, 2003c) in KDS, and discuss protection methods. In conventional database systems, data confidentiality is achieved by hiding sensitive data from unauthorized users (e.g. Data encryption or Access Control). However, hiding is not sufficient in KDS due to Chase. Chase is a generalized null value imputation algorithm that is designed to predict null or missing values, and has many application areas. For example, we can use Chase in a medical decision support system to handle difficult medical situations (e.g. dangerous invasive medical test for the patients who cannot take it). The results derived from the decision support system can help doctors diagnose and treat patients. The data approximated by Chase is particularly reliable because they reflect the actual characteristics of the data set in the information system.

Chase, however, can create data security problems if an information system contains confidential data (Im and Ras, 2005) (Im, 2006). Suppose that an attribute in an information system S contains medical information about patients; some portions of the data are not confidential while others have to be confidential. In this case, part or all of the confidential data in the attribute can be revealed by Chase using knowledge extracted at S. In other words, self-generated rules extracted from non-confidential portions of data can be used to find secret data.

Knowledge is often extracted from remote sites in a Distributed Knowledge Discovery System (DKDS) (Ras, 1994). The key concept of DKDS is to generate global knowledge through knowledge sharing. Each site in DKDS develops knowledge independently, and they are used jointly to produce global knowledge without complex data integrations. Assume that two sites S_1 and S_2 in a DKDS accept the same ontology of their attributes, and they share their knowledge in order to obtain global knowledge, and an attribute of a site S_1 in a DKDS is confidential. The confidential data in S_1 can be hidden by replacing them with null values. However, users at S_1 may treat them as missing data and reconstruct them with Chase using the knowledge extracted from S_2. A distributed medical information system is an example that an attribute is confidential for one information system while the same attribute may not be considered as secret information in another site. These examples show that hiding confidential data from an information system does not guarantee data confidentiality due to Chase, and methods that would protect against these problems are essential to build a security-aware KDS.

BACKGROUND

Data Security and Knowledge Discovery System

Security in KDS has been studied in various disciplines such as cryptography, statistics, and data mining. A well known security problem in cryptography area is how to acquire global knowledge in a distributed system while exchanging data securely. In other words, the objective is to extract global knowledge without disclosing any data stored in each local site. Proposed solutions are based primarily on the idea of secure multiparty protocol (Yao, 1996) that ensures each participant cannot learn more than its own input and outcome of

a public function. Various authors expanded the idea to build a secure data mining systems. Clifton and Kantarcioglou employed the concept to association rule mining for vertically and horizontally partitioned data (Kantarcioglou and Clifton, 2002). Du et al, (Du and Zhan, 2002) and Lindell et al, (Lindell and Pinkas, 2000) used the protocol to build a decision tree. They focused on improving the generic secure multiparty protocol for ID3 algorithm [Quinlan, 1993]. All these works have a common drawback that they require expensive encryption and decryption mechanisms. Considering that real world system often contain extremely large amount of data, performance has to be improved before we apply these algorithms. Another research area of data security in data mining is called perturbation. Dataset is perturbed (e.g. noise addition or data swapping) before its release to the public to minimize disclosure risk of confidential data, while maintaining statistical characteristics (e.g. mean and variable). Muralidhar and Sarathy (Muralidhar and Sarathy, 2003) provided a theoretical basis for data perturbation in terms of data utilization and disclosure risks. In KDD area, protection of sensitive rules with minimum side effect has been discussed by several researchers. In (Oliveira & Zaiane, 2002), authors suggested a solution to protecting sensitive association rules in the form of "sanitization process" where protection is achieved by hiding selective patterns from the frequent itemsets. There has been another interesting proposal (Saygin & Verykios & Elmagarmid, 2002) for hiding sensitive association rules. They introduced an interval of minimum support and confidence value to measure the degree of sensitive rules. The interval is specified by the user and only the rules within the interval are to be removed. In this article, we focus on data security problems in distributed knowledge sharing systems. Related works concentrated only on a standalone information system, or did not consider knowledge sharing techniques to acquire global knowledge.

Chase Algorithm

The overall steps for Chase algorithm is the following.

1. Identify all incomplete attribute values in S.
2. Extract rules from S describing these incomplete attribute values.
3. Null values in S are replaced by values (with their weights) suggested by the rules.
4. Steps 1-3 are repeated until a fixed point is reached.

More specifically, suppose that we have an incomplete information system $S = (X,A,V)$ where X is a finite set of object, A is a finite set of attributes, and V is a finite set of their values. Incomplete information system is a generalization of an information system introduced by (Pawlak, 1991). It is understood by having a set of weighted attribute values as a value of an attribute. In other words, multiple values can be assigned as an attribute value for an object with their weights (w). Assuming that a knowledge base $KB = \{(t \rightarrow v_c) \in D : c \in In(A)\}$ is a set of all classification rules extracted from S by $ERID(S, \lambda_1 \lambda_2)$, where $In(A)$ is the set of incomplete attributes in S, v_c is a value of attribute c, and λ_1, λ_2 are thresholds for minimum support and minimum confidence, correspondingly. *ERID* (Dardzinska and Ras, 2003b) is the algorithm for discovering rules from incomplete information systems, which can handle weighted attribute values. Assuming further that $Rs(x_i) \subseteq KB$ is the set of rules that all of the conditional part of the rules match with the attribute values in $x_i \in S$, and $d(x_i)$ is a null value, then, there are three cases for null value imputations (Dardzinska and Ras, 2003a, 2003c):

1. $Rs(x_i) = \Phi$. $d(x_i)$ cannot be replaced.
2. $Rs(x_i) = \{ r_1 = [t_1 \rightarrow d_1], r2 = [t_1 \rightarrow d_1], \ldots r_k = [t_k \rightarrow d_i]\}$. $d(x_i) = d_1$ because every rule predicts a single decision attribute value.
3. $Rs(x_i) = \{ r_1 = [t1 \rightarrow d_1], r2 = [t_1 \rightarrow d_2], \ldots r_k = [t_k \rightarrow d_k]\}$. Multiple values can replace $d(x_i)$.

Clearly, the weights of predicted values, which represent the strength of prediction, are 1 for case 2. For case 3, weight is calculated based on the confidence and support of rules used by Chase (Ras and Dardzinska, 2005b). Chase is an iterative process. An execution of the algorithm for all attributes in S typically generates a new information system, and the execution is repeated until it reaches a state where no improvement is achieved.

MAIN FOCUS OF THE ARTICLE

Reconstruction of Confidential Data by Chase

Suppose that an information system *S* is part of a knowledge discovery system (either single or distributed). Then, there are two cases in terms of the source of knowledge.

1. Knowledge is extracted from local site, *S*
2. Knowledge is extracted from remote site, S_i, *for* $S_i \neq S$

Let's first consider a simple example that illustrates how locally extracted rules can be used to reveal confidential data. Suppose that part of an attribute *d* is confidential and it is denoted as $v_{conf} = \{d(x_3), d(x_4)\}$. To protect v_{conf}, we hide them by constructing a new information system S_d, where

1. $a_{S(x)} = a_{S_d(x)}$, for any $a \in A - \{d\}, x \in X$
2. $d_{S_d(x)} = d_{S_d(x)} - v_{conf}$,

Now, we extract rules from S_d in terms of *d*. Assuming that a rule, $r_1 = a_1 \rightarrow d_3$ is extracted. It is applicable to objects that d_3 was hidden, meaning the conditional part of r_1 overlaps with the attribute value in the object. We can use Chase to predict the confidential data d_3. Clearly, it is possible that predicted values are not equal to the actual values or its weight is low. In general, there are three different cases.

1. $d_{S_d(x)} = d_{S(x)}$ and $w \geq \lambda$
2. $d_{S_d(x)} = d_{S(x)}$ and $w < \lambda$
3. $d_{S_d(x)} \neq d_{S(x)}$

where λ is the minimum threshold value for an information system (Ras and Dardzinska, 2005). Clearly, $d_{S_d(x)}$ in case 1 is our major concern because the confidence of approximated data is higher than λ. We do not need to take any action for case 2 and case 3.

The notion of confidential data disclosure by Chase can be extended to Distributed Knowledge Discovery System. The principal of DKDS is that each site develops knowledge independently, and the knowledge is used jointly to produce global knowledge (Ras, 1994), so that each site acquires global knowledge without implementing complex data integrations. The security problem in this environment is created by the knowledge extracted from remote sites. For example, assume that an attribute *d* in an information system *S* (See Table 1) is confidential and we hide *d* from *S* and construct $S_d = (X,A,V)$, where:

1. $a_{S(x)} = a_{S_d(x)}$, for any $a \in A - \{d\}, x \in X$
2. $d_{S_d(x)}$ is undefined, for any $x \in X$,

In this scenario, there exists no local rule describing *d* because *d* is completely hidden. Instead, rules are extracted from remote sites (e.g. r_1,r_2 in Table 2). Now, the process of missing value reconstruction is similar to that of local Chase. For example, $r_1 = b_1 \rightarrow d_1$ supports objects $\{x_1,x_3\}$, and $r_2 = a_2 \cdot b_2 \rightarrow d_2$ supports objects $\{x_4\}$. The confidential data, d_1, and d_2, can be reconstructed using these two rules.

Rules are extracted from different information systems in DKDS. Inconsistencies in semantics (if exists) have to be resolved before any null value imputation can be applied (Ras and Dardzinska, 2004a). In general, we assume that information stored in an ontology of a system (Guarino, and Giaretta, 1995),(Sowa, 1999, 2000) and in inter-ontologies among systems (if they are required and provided) are sufficient to resolve

Table 1. Information system S_d

X	a	B	c	d
x_1	a_1	b_1	c_1	hidden
x_2	a_1	b_2	c_1	hidden
x_3	a_2	b_1	c_3	hidden
x_4	a_2	b_2	c_2	hidden
x_5	a_3	b_2	c_2	hidden
x_6	a_3	b_2	c_4	hidden

Table 2. Rules in Knowledge Base

Rid	Rule	Support	confidence	Source
r_1	$b_1 \rightarrow d_1$	20%	100%	Remote
r_2	$a_2 \cdot b_2 \rightarrow d_2$	20%	100%	Remote
r_3	$c_1 \rightarrow a_1$	33%	100%	Local

inconsistencies in semantics of all sites involved in Chase.

Achieving Data Confidentiality

Clearly, additional data have to be hidden or modified to avoid reconstruction of confidential data by Chase. In particular, we need to change data from non-confidential part of the information system. An important issue in this approach is how to minimize data loss in order to preserve original data set as much as possible. The search space for finding the minimum data set is, in general, very large because of the large number of predictions made by Chase. In addition, there are multiple ways to hide data for each prediction. Several algorithms have been proposed to improve performance based on the discovery of Attribute Value Overlap (Im and Ras, 2005) and Chase Closure (Im , Ras and Dardzinska 2005a).

We can minimize data loss further by taking advantage of hierarchical attribute structure (Im , Ras and Dardzinska, 2005b). Unlike single-level attribute system, data collected with different granularity levels are assigned to an information system with their semantic relations. For example, when the age of Alice is recorded, its value can be either 20 or *young*. Assuming that exact age is sensitive and confidential, we may show her age as 'young' if revealing the value, *'young',* does not compromise her privacy. Clearly, such system provides more data to users compared to the system that has to hide data completely.

Unlike the previous example where knowledge is extracted and stored in KB before we apply a protection algorithm, some systems need to generate rules after we hide a set of data from an information system. In this case, we need to consider knowledge loss (in the form of rules). Now, the objective is that secrecy of data is being maintained while the loss of knowledge in each information systems is minimized (Ras and Dardzinska and Im, 2006). Clearly, as we start hiding additional attribute values from an information system S, we also start losing some knowledge because the data set may be different. One of the important measurements of knowledge loss can be its interestingness. The interestingness of knowledge (Silberschatz and Tuzhilin, 1996) is classified largely into two categories (Silberschatz and Tuzhilin, 1996): subjective and objective. Subjective measures are user-driven, domain-dependent. This type of measurement includes unexpectedness, novelty, actionable (Ras and Tsay, 2005) (Ras and Tsay, 2003) rules. Objective measures are data-driven and domain-independent. They evaluate rules based on statistics and structures of patterns, (e.g., support, confidence, etc).

If the KDS allows for users to use Chase with the rules generated with any support and confidence value, some of the confidential data protected by the described methods may be disclosed. This is obvious because Chase does not restrict minimum support and confidence of rules when it reconstructs null values. A naive solution to this problem is to run the algorithm with a large number of rules generated with wide range of confidence and support values. However, as we increase the size of KB, more attribute values will most likely have to be hidden. In addition, malicious users may use even lower values for rule extraction attributes, and we may end up with hiding all data. In fact, ensuring data confidentiality against all possible rules is difficult because Chase does not enforce minimum support and confidence of rules when it reconstructs missing data. Therefore, in these knowledge discovery systems, the security against Chase should aim to reduce the confidence of the reconstructed values, particularly, by meaningful rules, such as rules with high support or high confidence, instead of trying to prevent data reconstruction by all possible rules. One of the ways to protect confidential data in this environment is to find object reducts (Skowron and Rauszer 1992) and use

the reducts to remove attribute values that will more likely be used to predict confidential attribute values with high confidence.

FUTURE TRENDS

More knowledge discovery systems will use Chase as a key tool because Chase provides robust prediction of missing or unknown attribute values in knowledge discovery systems. Therefore, further research and development for data security and Chase (or privacy in data mining in general) will be conducted. This includes providing data protection algorithms for dynamic information system or improving usability of the methods for knowledge experts.

CONCLUSION

Hiding confidential data from an information system is not sufficient to provide data confidentiality against Chase in KDS. In this article, we presented the process of confidential data reconstruction by Chase, and solutions to reduce the risk. Additional data have to be hidden or modified to ensure the safekeeping of the confidential data from Chase. Data and knowledge loss have to be considered to minimize the changes to existing data and knowledge. If the set of knowledge used to Chase is not completely known, we need to hide data that will more likely be used to predict confidential data with high confidence

REFERENCES

Du, W. and Zhan, Z. (2002). Building decision tree classifier on private data. *Proceedings of the IEEE ICDM Workshop on Privacy, Security and Data Mining*.

Dardzinska, A., Ras, Z. (2003a). Chasing Unknown Values in Incomplete Information Systems, *Proceedings of ICDM 03 Workshop on Foundations and New Directions of Data Mining*.

Dardzinska, A., Ras, Z. (2003b). On Rules Discovery from Incomplete Information Systems, *Proceedings of ICDM 03 Workshop on Foundations and New Directions of Data Mining*.

Dardzinska, A., Ras, Z. (2003c). Rule-Based Chase Algorithm for Partially Incomplete Information Systems, *Proceedings of the Second International Workshop on Active Mining*.

Du, W. and Atallah, M. J. (2001). Secure multi-party computation problems and their applications: A review and open problems. *New Security Paradigms Workshop*

Guarino, N., Giaretta, P. (1995). Ontologies and knowledge bases, towards a terminological clarification, *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing*.

Im, S. (2006). Privacy aware data management and chase. *Fundamenta Informaticae, Special issue on intelligent information systems*. IOS Press.

Im, S., Ras, Z. (2005). Ensuring Data Security against Knowledge Discovery in Distributed Information System, *Proceedings of the 10th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing*.

Im, S., Ras, Z., Dardzinska, A. (2005a). Building A Security-Aware Query Answering System Based On Hierarchical Data Masking, *Proceedings of the ICDM Workshop on Computational Intelligence in Data Mining*.

Im, S., Ras, Z., Dardzinska, A. (2005b). SCIKD: Safeguarding Classified Information against Knowledge Discovery, *Proceedings of the ICDM Workshop on Foundations of Data Mining*

Kantarcioglou, M. and Clifton, C. (2002). Privacy-preserving distributed mining of association rules on horizontally partitioned data. *Proceedings of the ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, page 24-31.

Lindell, Y. and Pinkas, B. (2000). Privacy preserving data mining. *Proceedings of the 20th Annual International Cryptology Conference on Advances in Cryptology*, page 36-54.

Muralidhar, K. and Sarathy, R. (1999). Security of random data perturbation methods. *ACM Trans. Database System*, 24(4), page 487-493.

Oliveira, S. R. M. and Zaiane, O. R. (2002). Privacy preserving frequent itemset mining. *Proceedings of the*

IEEE ICDM Workshop on Privacy, Security and Data Mining, page 43-54.

Pawlak, Z. (1991). Rough sets-theoretical aspects of reasoning about data, Kluwer

Quinlan, J. (1993). C4.5: Programs for machine learning.

Ras, Z. (1994). Dictionaries in a distributed knowledge-based system, *Concurrent Engineering: Research and Applications, Concurrent Technologies Corporation*

Ras, Z., Dardzinska, A. (2004a). Ontology Based Distributed Autonomous Knowledge Systems, *Information Systems International* Journal, 29(1), page 47–58.

Ras, Z., Dardzinska, A. (2005). CHASE-2: Rule based chase algorithm for information systems of type lambda, *Proceedings of the Second International Workshop on Active Mining*

Ras Z., Dardzinska, A. (2005b). Data security and null value imputation in distributed information systems. *Advances in Soft Computing*, page 133-146. Springer-Verlag.

Zbigniew Ras, A. Dardzinska, Seunghyun Im, (2006). Data Security versus Knowledge Loss, *Proceedings of the International Conference on AI*

Ras, Z. and Tsay, L. (2005). Action rules discovery: System dear2, method and experiments. *Experimental and Theoretical Artificial Intelligence*

Ras, Z. and Tsay, L.-S. (2003). Discovering extended action rules (system dear). *Proceedings of intelligent information systems*, pages 293~300.

Saygin, Y., Verykios, V., and Elmagarmid, A. (2002). Privacy preserving association rule mining. *Proceedings of the 12th International Workshop on Research Issues in Data Engineering*, page 151~158.

Skowron A and Rauszer C. (1992). The discernibility matrices and functions in information systems. Intelligent Decision Support, 11:331–362.

Silberschatz, A., Tuzhilin, A. (1996). What makes patterns interesting in knowledge discovery systems. *IEEE Trans. On Knowledge And Data Engineering*, 8:970-974.

Sowa, J.F. (1999). Ontological categories, in L. Albertazzi, ed., Shapes of Forms: From Gestalt Psychology and Phenomenology to Ontology and Mathematics, Kluwer, 307-340.

Sowa, J.F. (2000). Knowledge Representation: Logical, Philosophical, and Computational Foundations, Brooks/Cole Publishing Co., Pacific Grove, CA.

Yao, A. C. (1996). How to generate and exchange secrets. *Proceedings of the 27th IEEE Symposium on Founda- tions of Computer Science*, pages 162~167.

KEY TERMS

Chase: A recursive strategy applied to a database V, based on functional dependencies or rules extracted from V, by which a null value or an incomplete value in V is replaced by more complete values.

Distributed Chase: A recursive strategy applied to a database V, based on functional dependencies or rules extracted both from V and other autonomous databases, by which a null value or an incomplete value in V is replaced by more complete values. Any differences in semantics among attributes in the involved databases have to be resolved first.

Data Confidentiality: Secrecy of confidential data by hiding a set of attribute values from unauthorized users in the knowledge discovery system.

Knowledge Base: A collection of rules defined as expressions written in predicate calculus. These rules have a form of associations between conjuncts of values of attributes.

Knowledge Discovery System: A set of information systems that is designed to extract and provide patterns and rules hidden from a large quantity of data. The system can be standalone or distributed.

Ontology: An explicit formal specification of how to represent objects, concepts and other entities that are assumed to exist in some area of interest and relationships holding among them. Systems that share the same ontology are able to communicate about domain of discourse without necessarily operating on a globally shared theory. System commits to ontology if its observable actions are consistent with the definitions in the ontology.

Data Cube Compression Techniques: A Theoretical Review

Alfredo Cuzzocrea
University of Calabria, Italy

INTRODUCTION

OnLine Analytical Processing (OLAP) research issues (Gray, Chaudhuri, Bosworth, Layman, Reichart & Venkatrao, 1997) such as data cube modeling, representation, indexing and management have traditionally attracted a lot of attention from the Data Warehousing research community. In this respect, as a fundamental issue, the problem of *efficiently compressing the data cube* plays a leading role, and it has involved a vibrant research activity in the academic as well as industrial world during the last fifteen years.

Basically, this problem consists in dealing with massive-in-size data cubes that make access and query evaluation costs prohibitive. The widely accepted solution consists in generating a *compressed representation* of the target data cube, with the goal of reducing its size (given an input space bound) while admitting loss of information and approximation that are considered irrelevant for OLAP analysis goals (e.g., see (Cuzzocrea, 2005a)). Compressed representations are also referred-in-literature under the term "*synopsis data structures*", i.e. succinct representations of original data cubes introducing a limited loss of information. Benefits deriving from the data cube compression approach can be summarized in a relevant reduction of computational overheads needed to both represent the data cube and evaluate resource-intensive OLAP queries, which constitute a wide query class allowing us to extract useful knowledge from huge amounts of multidimensional data repositories in the vest of summarized information (e.g., *aggregate information* based on popular SQL aggregate operators such as SUM, COUNT, AVG etc), otherwise infeasible by means of traditional OLTP approaches. Among such queries, we recall: (*i*) *range-queries*, which extract a sub-cube bounded by a given range; (*ii*) *top-k queries*, which extract the k data cells (such that k is an input parameter) having the highest aggregate values; (*iii*) *iceberg queries*, which extract the data cells having aggregate values above a given threshold.

This evidence has given raise to the proliferation of a number of *Approximate Query Answering* (AQA) techniques, which, based on data cube compression, aim at providing *approximate answers* to resource-intensive OLAP queries instead of computing exact answers, as decimal precision is usually negligible in OLAP query and report activities (e.g., see (Cuzzocrea, 2005a)).

Starting from these considerations, in this article we first provide a comprehensive, rigorous survey of main data cube compression techniques, i.e. histograms, wavelets, and sampling. Then, we complete our analytical contribution with a detailed theoretical review focused on spatio-temporal complexities of these techniques.

BACKGROUND

A *data cube* $\mathcal{L}$ is a tuple $\mathcal{L} = \langle C,\mathcal{J},\mathcal{H},\mathcal{M}\rangle$, such that: (*i*) C is the data domain of $\mathcal{L}$ containing (OLAP) *data cells*, which are the basic SQL aggregations of $\mathcal{L}$ computed against the relational data source S alimenting $\mathcal{L}$; (*ii*) $\mathcal{J}$ is the set of *dimensions* of $\mathcal{L}$, i.e. the *functional attributes* (of S) with respect to which the underlying OLAP analysis is defined (in other words, $\mathcal{J}$ is the set of attributes with respect to which relational tuples in S are aggregated); (*iii*) $\mathcal{H}$ is the set of *hierarchies* related to the dimensions of $\mathcal{L}$, i.e. hierarchical representations of the functional attributes shaped-in-the-form-of generic trees; (*iv*) $\mathcal{M}$ is the set of *measures* of $\mathcal{L}$, i.e. the *attributes of interest* (of S) for the underlying OLAP analysis (in other words, $\mathcal{M}$ is the set of attributes with respect to which SQL aggregations stored in data cells of $\mathcal{L}$ are computed).

HISTOGRAMS

Histograms have been extensively studied and applied in the context of *Selectivity Estimation* (Kooi, 1980), and are effectively implemented in commercial systems (e.g., Oracle Database, IBM DB2 Universal Database, Microsoft SQL Server) to query optimization purposes. In statistical databases (Shoshani, 1997), histograms represent a method for approximating probability distributions. They have also been used in data mining activities, intrusion detection systems, scientific databases, i.e. in all those applications which (*i*) operate on huge numbers of detailed records, (*ii*) extract useful knowledge only from condensed information consisting of summary data, (*iii*) but are not usually concerned with detailed information. Indeed, histograms can reach a surprising efficiency and effectiveness in approximating the actual distributions of data starting from summarized information. This has led the research community to investigate the use of histograms in the fields of *DataBase Management Systems* (DBMS) (Kooi, 1980; Poosala & Ioannidis, 1997; Ioannidis & Poosala, 1999; Acharya, Poosala & Ramaswamy, 1999; Gunopulos, Kollios, Tsotras & Domeniconi, 2000; Bruno, Chaudhuri & Gravano, 2001), and *OnLine Analytical Processing* (OLAP) systems (Poosala & Ganti, 1999; Buccafurri, Furfaro, Saccà & Sirangelo, 2003; Cuzzocrea, 2005a; Cuzzocrea & Wang, 2007; Leng, Bao, Wang & Yu, 2007).

Histogram literature has a long history, what confirms the prominence of histogram research within the broader context of database/data-cube compression techniques. They are data structures obtained by partitioning a data distribution (or, equally, a data domain) into a number of mutually disjoint blocks, called *buckets*, and then storing, for each bucket, some aggregate information of the corresponding range of values, like the sum of values in that range (i.e., applying the SQL aggregate operator SUM), or the number of occurrences (i.e., applying the SQL aggregate operator COUNT), such that this information retains a certain "summarizing content".

Histograms are widely used to support two kinds of applications: (*i*) selectivity estimation inside *Query Optimizers* of DBMS, as highlighted before, and (*ii*) *approximate query answering* against databases and data cubes. In the former case, the data distribution to be compressed consists of the frequencies of values of attributes in a relation (it should be noted that, in this vest, histograms are mainly used within the core layer of DBMS, thus dealing with databases properly). In the latter case, the data distribution to be compressed consists of the data items of the target domain (i.e., a database or a data cube) directly, and the goal is to provide fast and approximate answers to resource-intensive queries instead of waiting-for time-consuming exact evaluations of queries. They are a very-popular class of synopsis data structures, so that they have been extensively used in the context of approximate query answering techniques. Early experiences concerning this utilization of histograms are represented by the work of Ioannidis and Poosala (1999), that propose using histograms to provide approximate answers to set-valued queries, and the work of Poosala and Ganti (1999), that propose using histograms to provide approximate answers to range-queries in OLAP.

There are several classes of histograms in literature. We distinguish between *one-dimensional histograms*, i.e. histograms devoted to compress one-dimensional data domains, and *multidimensional histograms*, i.e. histograms working on multidimensional data domains, which are more interesting than the former, and can be found in a wide range of modern, large-scale, *data-intensive* applications. Moreover, we can also further distinguish between *static histograms* and *dynamic histograms*. The first ones are statically computed against the target domain, and are not particularly suitable to efficiently accomplish data updates occurring on original data sources; the second ones are dynamically computed by taking into consideration, beyond the target domain, or, in some cases, a synopsis of it, other entities related to the dynamics of the target DBMS/OLAP server such as *query-workloads*, *query feedbacks*, *load balancing issues* etc. Contrarily to the previous histogram class, dynamic histograms efficiently support update management, being their partition dependent on a "parametric" configuration that can be easily (re-)computed at will.

With respect to our work, we are mainly interested in multidimensional histograms, since multidimensional data domains well-represent OLAP data cubes. A popular and conventional approach is based on the well-known *Attribute-Value Independence* (AVI) assumption, according to which any query involving a set of attributes can be answered by applying it on each attribute singularly. This approach is theoretically

reasonable, but it has been recognized as source of gross errors in practice (e.g., (Poosala & Ioannidis, 1997)). To cope with this problem, multidimensional histograms use a small number of multidimensional buckets to *directly* approximate the joint data distribution.

Static Histograms

Among all the alternatives, we focus our attention on the following static multidimensional histograms, mainly because they can be considered as representative and significant experiences in this context: *Min-Skew* (Acharya, Poosala & Ramaswamy, 1999) and *GenHist* (Gunopulos, Kollios, Tsotras & Domeniconi, 2000) histograms.

Min-Skew histogram was originally designed in (Acharya, Poosala & Ramaswamy, 1999) to tackle the problem of selectivity estimation of *spatial data* in *Geographical Information Systems* (GIS). Spatial data are referred to *spatial* (or *geographical*) *entities* such as points, lines, poly-lines, polygons and surfaces, and are very often treated by means of minimal rectangles containing them, namely *Minimum Bounding Rectangles* (MBR). *Min-Skew* is more sophisticated than *MHist*. The main idea behind a *Min-Skew* histogram $H_{M\text{-}S}(D)$ is to follow the criterion of minimizing the *spatial skew* of the histogram by performing a *Binary Space Partitioning* (BSP) via recursively dividing the space along one of the dimensions each time. More formally, each point in the space of a given GIS instance is associated to a *spatial density*, defined as the number of MBR that contain such a point.

(Gunopulos, Kollios, Tsotras, & Domeniconi, 2000) proposes *GenHist histogram*, a new kind of multidimensional histogram that is different from the previous ones with respect to the build procedure. The key idea is the following: given an histogram H whit h_b buckets on an input data domain D, a *GenHist* histogram $H_{GH}(D)$ is built by finding n_b overlapping buckets on H, such that n_b is an input parameter. To this end, the technique individuates the number of distinct regions that is much larger than the original number of buckets h_b, thanks to a greedy algorithm that considers *increasingly-coarser grids*. At each step, such algorithm selects the set of cells J of highest density, and moves enough randomly-selected points from J into a bucket to make J and its neighbors "close-to-uniform".

Dynamic Histograms

Dynamic multidimensional histograms extend capabilities of static multidimensional histograms by incorporating inside their generating algorithms the amenity of building/refining the underlying partition in dependence on non-conventional entities related to the dynamic behavior of the target DBMS/OLAP server, such as query-workloads. Among this class of histograms, relevant proposal are: *STHoles* (Bruno, Chaudhuri & Gravano, 2001) histograms and the innovative data structure **TP**-*Tree* (Cuzzocrea & Wang, 2007).

(Bruno, Chaudhuri & Gravano, 2001) proposes a different kind of multidimensional histogram, based on the analysis of the query-workload on it: the *workload-aware histogram*, which they call *STHoles*. Rather than an arbitrary overlap, a *STHoles* histogram $H_{ST}(D)$ allows bucket nesting, thus achieving the definition of the so-called *bucket tree*. Query-workloads are handled as follows: the query result stream Q^R to be analyzed is intercepted and, for each query Q_j belonging to Q^R and for each bucket b_i belonging to the current bucket tree, the number $|Q_j \cap b_i|$ is counted. Then, "holes" in b_i for regions of different *tuple density* are "drilled" and "pulled out" as children of b_i. Finally, buckets of similar densities are merged in such a way as to keep the number of buckets constant.

T*unable*-**P***artition*-*Tree* (**TP**-*Tree*) (Cuzzocrea & Wang, 2007) is a tree-like, highly-dynamic data structure that codifies a multidimensional histogram for massive (multidimensional) data cubes, denoted by $H_{TP}(D)$, whose partition *varies over time* according to the query-workload against the target OLAP server. For this reason, partitions of $H_{TP}(D)$ are named as *tunable partitions*, and $H_{TP}(D)$ is said to be a "workload-aware" synopsis data structure. The main contribution of the **TP**-*Tree* proposal with respect to previous techniques consists in introducing models and algorithms having low computational costs, whereas previous techniques are, usually, time-consuming and resource-intensive. Data stored inside buckets of **TP**-*Tree* partitions are obtained by (*i*) *sampling* the input data cube (from this solution, low computational costs required by the **TP**-*Tree* approach follow), and (*ii*) separately representing, storing, and indexing *outliers* via high performance *quad-tree* based (data) structures. Also, **TP**-*Tree* is able to provide *probabilistic guarantees* over the degree of approximation of the answers, which is a leading

research topic very often neglected by the research community (e.g., see (Cuzzocrea, 2005b)).

WAVELETS

Wavelets (Stollnitz, Derose & Salesin, 1996) are a mathematical transformation which defines a hierarchical decomposition of functions (representing signals or data distributions) into a set of coefficients, called *wavelet coefficients*. In more detail, wavelets represent a function in terms of a coarse overall shape, plus details that range from coarse to fine. They were originally applied in the field of image and signal processing. Recent studies have shown the applicability of wavelets to selectivity estimation, as well as to the approximation of both specific forms of query (like range-queries) (Vitter, Wang & Iyer, 1998), and "general" queries (Chakrabarti, Garofalakis, Rastogi & Shim, 2000) (using join operators) over data cubes.

Specifically, the compressed representation of data cubes via wavelets (Vitter, Wang & Iyer, 1998) is obtained in two steps. First, a wavelet transform is applied to the data cube, thus generating a sequence of coefficients. At this step no compression is obtained (the number of wavelet coefficients is the same as the number of data points in the examined distribution), and no approximation is introduced, as the original data distribution can be reconstructed exactly applying the inverse of the wavelet transform to the sequence of coefficients. Next, among the N wavelet coefficients, only the $m << N$ most "significant" ones are retained, whereas the others are "thrown away", and their value is implicitly set to zero. The set of retained coefficients defines the compressed representation, called *wavelet synopses*, which refer to synopses computed by means of wavelet transformations. This selection is driven by a some criterion, which determines the lost of information that introduces approximation.

SAMPLING

Random sampling-based methods propose mapping the original multidimensional data domain in a smaller subset by sampling: this allows a more compact representation of the original data to be achieved. Query performances can be significantly improved by pushing sampling to Query Engines, with computational overheads that are very low when compared to more resource-intensive techniques such as multidimensional histograms and wavelet decomposition. (Hellerstein, Haas & Wang, 1997) proposes a system for effectively supporting online aggregate query answering, and also providing probabilistic guarantees about the accuracy of the answers in terms of *confidence intervals*. Such system allows a user to execute an aggregate query and to observe its execution in a graphical interface that shows both the partial answer and the corresponding (partial) confidence interval. The user can also stop the query execution when the answer has achieved the desired degree of approximation. No synopses are maintained since the system is based on a random sampling of the tuples involved in the query. Random sampling allows an *unbiased estimator* for the answer to be built, and the associated confidence interval is computed by means of the Hoeffding's inequality. The drawback of this proposal is that response time needed to compute answers can increase since sampling is done at query time. The absence of synopses ensures that there are not additional computational overheads because no maintenance tasks must be performed.

(Gibbons & Matias, 1998) propose the idea of using sampling-based techniques to tame spatio-temporal computational requirements needed for evaluating resource-intensive queries against massive data warehouses (e.g., OLAP queries). Specifically, they argue to build synopses via sampling, such that these synopses provide a *statistical description* of data they model. Synopses are built in dependence on the specific class of queries they process (e.g., based on popular random sampling, or more complex sampling schemes), and are stored in the target data warehouse directly. (Acharya, Gibbons, Poosala & Ramaswamy, 1999) extends this approach as to deal with more interesting data warehouse queries such as join queries, which extract correlated data/knowledge from multiple fact tables. To this end, authors propose using a *pre-computed* sample synopses that considers samples of a small set of distinguished joins rather than accessing all the data items involved by the actual join. Unfortunately, this method works well only for queries with foreign-key joins which are known beforehand, and does not support arbitrary join queries over arbitrary schema.

THEORETICAL ANALYSIS AND RESULTS

In order to complement the actual contribution of the Data Warehousing research community to the data cube compression problem, in this Section we provide a theoretical analysis of main approaches present in literature, and derive results in terms of complexities of all the phases of data cube management activities correlated to such problem, e.g. computing/updating the compressed representation of the data cube and evaluating approximate queries against the compressed representation. First, we formally model the complexities which are at the basis of the theoretical analysis we provide. These complexities are:

- *Temporal complexity* of computing/updating the compressed representation of the data cube, denoted by $C_{\{C,U\}}[t]$;
- *Spatial complexity* of the compressed representation of the data cube, denoted by $C_{\{C,U\}}[s]$;
- *Temporal complexity* of evaluating approximate queries against the compressed representation of the data cube, denoted by $C_Q[t]$.

Being infeasible to derive closed formulas of spatio-temporal complexities such that these formulas are valid for all the data cube compression techniques presented in this paper (as each technique makes use of particular, specialized synopsis data structures and build, update and query algorithms), we introduce ad-hoc *complexity classes* modeling an *asymptotic analysis* of spatio-temporal complexities of the investigated techniques.

These complexity classes depend on (*i*) the size of input, denoted by *N*, which, without loss of generality, can be intended as the number of data cells of the original data cube, and (*ii*) the size of the compressed data structure computed on top of the data cube, denoted by *n*. Contrarily to *N*, the parameter *n* must be specialized for each technique class. For what regards histograms, *n* models the number of buckets; for wavelets, *n* models the number of number of wavelet coefficients; finally, for sampling, *n* models the number of sampled data cells.

To meaningfully support our theoretical analysis according to the guidelines given above, the complexity classes we introduce are:

- LOG(•), which models a logarithmic complexity with respect to the size of input;
- LINEAR(•), which models a linear complexity with respect to the size of input;
- POLY(•), which models a polynomial complexity with respect to the size of input.

It is a matter to note that the complexity modeled by these classes grows from LOG to POLY class. However, these classes are enough to cover all the cases of complexities of general data cube compression techniques, and to put in evidence similar and differences among such techniques.

Table 1 summarizes the results of our theoretical analysis. For what regards the temporal complexity of building/updating the compressed data cube, we observe that sampling-based techniques constitute the best case, whereas histograms and wavelets require a higher temporal complexity to accomplish the build/update task. For what regards the spatial complexity of the compressed data cube, we notice that all the techniques give us, at the asymptotic analysis, the same complexity, i.e. we approximately obtain the same benefits in terms of space occupancy from using any of such techniques. Finally, for what regards

Table 1. Theoretical analysis of data cube compression techniques

Technique	$C_{\{C,U\}}[t]$	$C_{\{C,U\}}[s]$	$C_Q[t]$
Histograms	POLY(N)	POLY(n)	LOG(n)
Wavelets	POLY(N)	POLY(n)	LOG(n)
Sampling	LINEAR(N)	POLY(n)	LINEAR(n)

temporal complexity of evaluating approximate queries against the compressed data cube, histograms and wavelets are better that sampling-based techniques. This because histograms and wavelets make use of specialized hierarchical data structures (e.g., bucket tree, decomposition tree etc) that are particularly efficient in supporting specialized kinds of OLAP queries, such as range-queries. This allows the query response time to be improved significantly.

FUTURE TRENDS

Future research efforts in the field of data cube compression techniques include the following themes: (*i*) ensuring *probabilistic guarantees* over the degree of approximate answers (e.g., (Garofalakis & Kumar, 2004; Cuzzocrea, 2005b; Cuzzocrea & Wang, 2007)), which deals with the problem of how to retrieve approximate answers that are probabilistically-close to exact answers (which are unknown at query time); (*ii*) devising *complex indexing schemes* for compressed OLAP data able to be *adaptive* in dependence on the query-workload of the target OLAP server, which plays a primary role in next-generation highly-scalable OLAP environments (e.g., those inspired to the novel publish/subscribe deployment paradigm).

CONCLUSION

Data-intensive research and technology have evolved from traditional relational databases to more recent data warehouses, and, with a prominent emphasis, to OLAP. In this context, the problem of efficiently compressing a data cube and providing approximate answers to resource-intensive OLAP queries plays a leading role, and, due to a great deal of attention from the research communities, a wide number of solutions dealing with such a problem have appeared in literature. These solutions are mainly focused to obtain an effective and efficient data cube compression, trying to mitigate computational overheads due to accessing and processing massive-in-size data cubes. Just like practical issues, theoretical issues are interesting as well, and, particularly, studying and rigorously formalizing spatio-temporal complexities of state-of-the-art data cube compression models and algorithms assumes a relevant role in order to meaningfully complete the contribution of these proposals.

REFERENCES

Acharya, S., Gibbons, P.B., Poosala, V., & Ramaswamy, S. (1999). Join Synopses for Approximate Query Answering. *Proceedings of the 1999 ACM International Conference on Management of Data*, 275-286.

Acharya, S., Poosala, V., & Ramaswamy, S. (1999). Selectivity Estimation in Spatial Databases. *Proceedings of the 1999 ACM International Conference on Management of Data*, 13-24.

Bruno, N., Chaudhuri, S., & Gravano, L. (2001). STHoles: A Multidimensional Workload-Aware Histogram. *Proceedings of the 2001 ACM International Conference on Management of Data*, 211-222.

Buccafurri, F., Furfaro, F., Saccà, D., & Sirangelo, C. (2003). A Quad-Tree based Multiresolution Approach for Two-Dimensional Summary Data. *Proceedings of the IEEE 15th International Conference on Scientific and Statistical Database Management*, 127-140.

Chakrabarti, K., Garofalakis, M., Rastogi, R., & Shim, K. (2000). Approximate Query Processing using Wavelets. *Proceedings of the 26th International Conference on Very Large Data Bases*, 111-122.

Cuzzocrea, A. (2005a). Overcoming Limitations of Approximate Query Answering in OLAP. *Proceedings of the 9th IEEE International Database Engineering and Applications Symposium*, 200-209.

Cuzzocrea, A. (2005b). Providing Probabilistically-Bounded Approximate Answers to Non-Holistic Aggregate Range Queries in OLAP. *Proceedings of the 8th ACM International Workshop on Data Warehousing and OLAP*, 97-106.

Cuzzocrea, A., & Wang, W. (2007). Approximate Range-Sum Query Answering on Data Cubes with Probabilistic Guarantees. *Journal of Intelligent Information Systems*, 28(2), 161-197.

Garofalakis, M.N., & Kumar, A. (2004). Deterministic Wavelet Thresholding for Maximum-Error Metrics", *Proceedings of the 23th ACM International Symposium on Principles of Database Systems*, 166-176.

Gibbons, P.B., & Matias, Y. (1998). New Sampling-Based Summary Statistics for Improving Approximate Query Answers. *Proceedings of the 1998 ACM International Conference on Management of Data*, 331-342.

Gray, J., Chaudhuri, S., Bosworth, A., Layman, A., Reichart, D., & Venkatrao, M. (1997). Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals. *Data Mining and Knowledge Discovery*, 1(1), 29-53.

Gunopulos, D., Kollios, G., Tsotras, V.J., & Domeniconi, C. (2000). Approximating Multi-Dimensional Aggregate Range Queries over Real Attributes. *Proceedings of the 2000 ACM International Conference on Management of Data*, 463-474.

Hellerstein, J.M., Haas, P.J., & Wang, H.J. (1997). Online Aggregation. *Proceedings of the 1997 ACM International Conference on Management of Data*, 171–182.

Ioannidis, Y., & Poosala, V. (1999). Histogram-based Approximation of Set-Valued Query Answers. *Proceedings of the 25th International Conference on Very Large Data Bases*, 174-185.

Kooi, R.P. (1980). *The Optimization of Queries in Relational Databases*. PhD Thesis, Case Western Reserve University.

Leng, F., Bao, Y., Wang, D., & Yu, G. (2007). A Clustered Dwarf Structure to Speed Up Queries on Data Cubes. *Proceedings of the 9th on Data Warehousing and Knowledge Discovery*, 170-180.

Poosala, V., & Ganti, V. (1999). Fast Approximate Answers to Aggregate Queries on a Data Cube. *Proceedings of the 11th IEEE International Conference on Statistical and Scientific Database Management*, 24-33.

Poosala, V., & Ioannidis, Y. (1997). Selectivity Estimation Without the Attribute Value Independence Assumption. *Proceedings of the 23th International Conference on Very Large Data Bases*, 486-495.

Shoshani, A. (1997). OLAP and Statistical Databases: Similarities and Differences. *Proceedings of the 16th ACM International Symposium on Principles of Database Systems*, 185-196.

Stollnitz, E.J., Derose, T.D., & Salesin, D.H. (1996). *Wavelets for Computer Graphics*. Morgan Kauffmann Publishers.

Vitter, J.S., Wang, M., & Iyer, B. (1998). Data Cube Approximation and Histograms via Wavelets. *Proceeding of the 7th ACM International Conference on Information and Knowledge Management*, 96-104.

KEY TERMS

OLAP Query: A query defined against a data cube that introduces a multidimensional range and a SQL aggregate operator, and returns as output the aggregate value computed over cells of the data cube contained in that range.

On-Line Analytical Processing (OLAP): A methodology for representing, managing and querying massive DW data according to multidimensional and multi-resolution abstractions of them.

On-Line Transaction Processing (OLTP): A methodology for representing, managing and querying DB data generated by user/application transactions according to flat (e.g., relational) schemes.

Relational Query: A query defined against a database that introduces some predicates over tuples stored in the database, and returns as output the collection of tuples satisfying those predicates.

Selectivity of a Relational Query: A property of a relational query that estimates the cost required to evaluate that query in terms of the number of tuples involved.

Selectivity of an OLAP Query: A property of an OLAP query that estimates the cost required to evaluate that query in terms of the number of data cells involved.

A Data Distribution View of Clustering Algorithms

Junjie Wu
Tsinghua University, China

Jian Chen
Tsinghua University, China

Hui Xiong
Rutgers University, USA

INTRODUCTION

Cluster analysis (Jain & Dubes, 1988) provides insight into the data by dividing the objects into groups (clusters), such that objects in a cluster are more similar to each other than objects in other clusters. Cluster analysis has long played an important role in a wide variety of fields, such as psychology, bioinformatics, pattern recognition, information retrieval, machine learning, and data mining. Many clustering algorithms, such as K-means and Unweighted Pair Group Method with Arithmetic Mean (UPGMA), have been well-established.

A recent research focus on clustering analysis is to understand the strength and weakness of various clustering algorithms with respect to data factors. Indeed, people have identified some data characteristics that may strongly affect clustering analysis including high dimensionality and sparseness, the large size, noise, types of attributes and data sets, and scales of attributes (Tan, Steinbach, & Kumar, 2005). However, further investigation is expected to reveal whether and how the data distributions can have the impact on the performance of clustering algorithms. Along this line, we study clustering algorithms by answering three questions:

1. What are the systematic differences between the distributions of the resultant clusters by different clustering algorithms?
2. How can the distribution of the "true" cluster sizes make impact on the performances of clustering algorithms?
3. How to choose an appropriate clustering algorithm in practice?

The answers to these questions can guide us for the better understanding and the use of clustering methods. This is noteworthy, since 1) in theory, people seldom realized that there are strong relationships between the clustering algorithms and the cluster size distributions, and 2) in practice, how to choose an appropriate clustering algorithm is still a challenging task, especially after an algorithm boom in data mining area. This chapter thus tries to fill this void initially. To this end, we carefully select two widely used categories of clustering algorithms, i.e., K-means and Agglomerative Hierarchical Clustering (AHC), as the representative algorithms for illustration. In the chapter, we first show that K-means tends to generate the clusters with a relatively uniform distribution on the cluster sizes. Then we demonstrate that UPGMA, one of the robust AHC methods, acts in an opposite way to K-means; that is, UPGMA tends to generate the clusters with high variation on the cluster sizes. Indeed, the experimental results indicate that the variations of the resultant cluster sizes by K-means and UPGMA, measured by the Coefficient of Variation (CV), are in the specific intervals, say [0.3, 1.0] and [1.0, 2.5] respectively. Finally, we put together K-means and UPGMA for a further comparison, and propose some rules for the better choice of the clustering schemes from the data distribution point of view.

BACKGROUND

People have investigated clustering algorithms from various perspectives. Many data factors, which may strongly affect the performances of clustering schemes, have been identified and addressed. Among them the

high dimensionality, the large size, and the existence of noise and outliers are typically the major concerns.

First, it has been well recognized that high dimensionality can make negative impact on various clustering algorithms which use Euclidean distance (Tan, Steinbach, & Kumar, 2005). To meet this challenge, one research direction is to make use of dimensionality reduction techniques, such as Multidimensional Scaling, Principal Component Analysis, and Singular Value Decomposition (Kent, Bibby, & Mardia, 2006). A detailed discussion on various dimensionality reduction techniques for document data sets has been provided by Tang et al. (2005). Another direction is to redefine the notions of proximity, e.g., by the Shared Nearest Neighbors similarity (Jarvis & Patrick, 1973). Some similarity measures, e.g., cosine, have also shown appealing effects on clustering document data sets (Zhao & Karypis, 2004).

Second, many clustering algorithms that work well for small or medium-size data sets are unable to handle large data sets. For instance, AHC is very expensive in terms of its computational and storage requirements. Along this line, a discussion of scaling K-means to large data sets was provided by Bradley et al. (1998). Also, Ghosh (2003) discussed the scalability of clustering methods in depth. A more broad discussion of specific clustering techniques can be found in the paper by Murtagh (2000). The representative techniques include CURE (Guha, Rastogi, & Shim, 1998), BIRCH (Zhang, Ramakrishnan, & Livny, 1996), CLARANS (Ng & Han, 2002), etc.

Third, outliers and noise in the data can also degrade the performance of clustering algorithms, such as K-means and AHC. To deal with this problem, one research direction is to incorporate some outlier removal techniques before conducting clustering. The representative techniques include LOF (Breunig, Kriegel, Ng, & Sander, 2000), HCleaner (Xiong, Pandey, Steinbach, & Kumar, 2006), etc. Another research direction is to handle outliers during the clustering process. There have been several techniques designed for such purpose, e.g., DBSCAN (Ester, Kriegel, Sander, & Xu, 1996), Chameleon (Karypis, Han, & Kumar, 1999), SNN density-based clustering (Ertoz, Steinbach, & Kumar, 2001), and CURE (Guha, Rastogi, & Shim, 1998).

In this chapter, however, we focus on understanding the impact of the data distribution, i.e., the distribution of the "true" cluster sizes, on the performances of K-means and AHC, which is a natural extension of our previous work (Xiong, Wu, & Chen, 2006; Wu, Xiong, Wu, & Chen, 2007). Also, we propose some useful rules for the better choice of clustering algorithms in practice.

MAIN FOCUS

Here, we explore the relationship between the data distribution and the clustering algorithms. Specifically, we first introduce the statistic, i.e., Coefficient of Variation (CV), to characterize the distribution of the cluster sizes. Then, we illustrate the effects of K-means clustering and AHC on the distribution of the cluster sizes, respectively. Finally, we compare the two effects and point out how to properly utilize the clustering algorithms on data sets with different "true" cluster distributions. Due to the complexity of this problem, we also conduct extensive experiments on data sets from different application domains. The results further verify our points.

A Measure of Data Dispersion Degree

Here we introduce the Coefficient of Variation (CV) (DeGroot & Schervish, 2001), which measures the dispersion degree of a data set. The CV is defined as the ratio of the standard deviation to the mean. Given a set of data objects $X = \{x_1, x_2,...,x_n\}$, we have $CV = s/\bar{x}$ where

$$\bar{x} = \sum_{i=1}^{n} x_i / n \text{ and}$$

$$s = \sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2 / (n-1)}.$$

Note that there are some other statistics, such as standard deviation and skewness (DeGroot & Schervish, 2001), which can also be used to characterize the dispersion of a data distribution. However, the standard deviation has no scalability; that is, the dispersion of the original data and stratified sample data is not equal if the standard deviation is used. Indeed, this does not agree with our intuition. Meanwhile, skewness cannot catch the dispersion in the situation that the data is symmetric but has high variance. In contrast, the CV is a dimensionless number that allows comparison of

the variation of populations that have significantly different mean values. In general, the larger the CV value is, the greater the variability is in the data.

In our study, we use CV to measure the dispersion degree of the cluster sizes before and after clustering. In particular, we use CV_0 to denote the "true" cluster distribution, i.e., the dispersion degree of the cluster sizes before clustering, and CV_1 to denote the resultant cluster distribution, i.e., the dispersion degree of the cluster sizes after clustering. And the difference of CV_1 and CV_0 is denoted by DCV, i.e., $DCV=CV_1-CV_0$.

The Uniform Effect of K-means

K-means is a prototype-based, simple partitional clustering technique which attempts to find a user-specified *K* number of clusters. These clusters are represented by their centroids (a cluster centroid is typically the mean of the points in the cluster). The clustering process of K-means is as follows. First, *K* initial centroids are selected, where *K* is specified by the user and indicates the desired number of clusters. Every point in the data is then assigned to the closest centroid, and each collection of points assigned to a centroid forms a cluster. The centroid of each cluster is then updated based on the points assigned to it. This process is repeated until no point changes clusters.

Figure 1 shows a sample data set with three "true" clusters. The numbers of points in Cluster 1, 2 and 3 are 96, 25 and 25, respectively. Thus the CV_0 value is 0.84. In this data, Cluster 2 is much closer to Cluster 1 than Cluster 3. Figure 2 shows the clustering results by K-means on this data set. One observation is that Cluster 1 is broken: part of Cluster 1 is merged with Cluster 2 as new Cluster 2 and the rest of cluster 1 forms new Cluster 1. However, the size distribution of the resulting two clusters is much more uniform now, i.e., the CV_1 value is 0.46. This is called the "uniform effect" of K-means on "true" clusters with different sizes. Another observation is that Cluster 3 is precisely identified by K-means. It is due to the fact that the objects in Cluster 3 are far away from Cluster 1. In other words, the uniform effect has been dominated by the large distance between two clusters. Nevertheless, our experimental results below show that the uniform effect of K-means clustering is dominant in most of applications, regardless the present of other data factors that may "mitigate" it.

The Dispersion Effect of UPGMA

The key component in agglomerative hierarchical clustering (AHC) algorithms is the similarity metric used to determine which pair of sub-clusters to be merged. In this chapter, we focus on the UPGMA (Unweighted Pair Group Method with Arithmetic mean) scheme which can be viewed as the tradeoff between the simplest single-link and complete-link schemes (Jain & Dubes, 1988). Indeed, UPGMA defines cluster similarity in terms of the average pair-wise similarity between the objects in the two clusters. UPGMA is widely used because it is more robust than many other agglomerative clustering approaches.

Figure 1. Clusters before k-means clustering

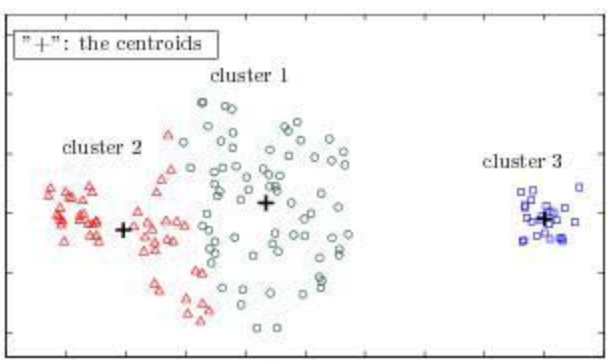


Figure 2. Clusters after k-means clustering

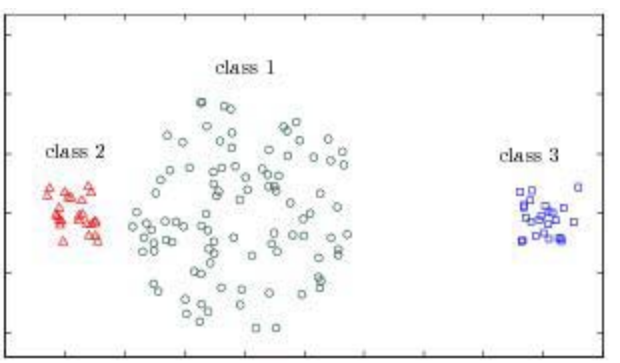


Figure 3. Clusters before employing UPGMA

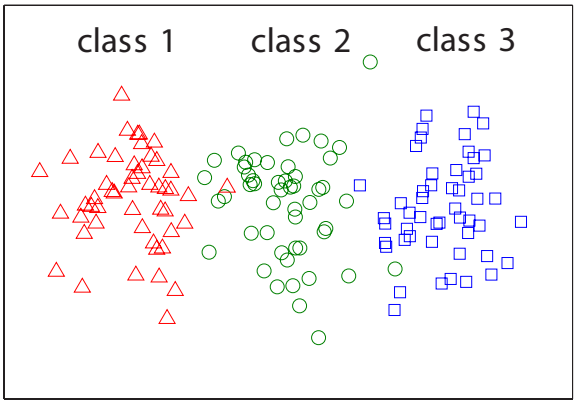


Figure 4. Clusters after employing UPGMA

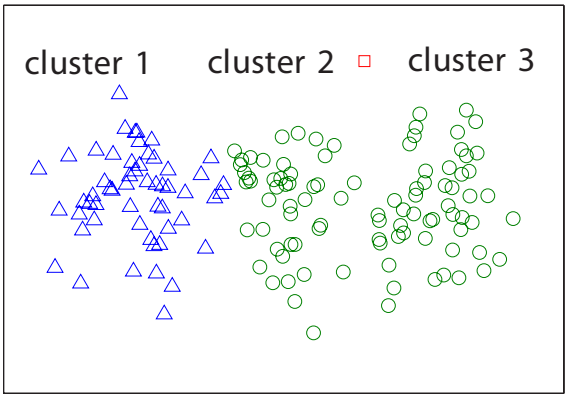


Figure 3 shows a sample data set with three "true" clusters, each of which contains 50 objects generated by one of three normal distributions. After exploiting UPGMA on this data set, we have three resultant clusters shown in Figure 4. Apparently, Class 3 is totally "disappeared", for all of its objects have been "absorbed" by the objects of Class 2. And the only one object farthest from the population forms Cluster 3. We can also capture this by the data distribution information. Actually, the CV_1 value of the resultant cluster sizes is 0.93, much higher than the CV_0 value of the "true" cluster sizes, i.e., 0. This is called the "dispersion effect" of UPGMA. From the above case, we can conclude that UPGMA tends to show the dispersion effect on data sets with noise or some small groups of objects that are far away from the populations. Considering the wide existences of noise or remote groups of objects in real-world data sets, we can expect that the dispersion effect of UPGMA holds in most of the cases. We leave this to the experiment results.

Experimental Results

Here we present experimental results to verify the different effects of K-means clustering and AHC on the resultant cluster sizes.

Experimental Setup

In the experiments, we used CLUTO implementations of K-means and UPGMA (Karypis, 2003). For all the experiments, the cosine similarity is used in the objective function. Other parameters are set with the default values.

For the experiments, we used a number of real-world data sets that were obtained from different application domains. Some characteristics of these data sets are shown in Table 1. Among the data sets, 12 out of 16 are the benchmark document data sets which were distributed with the CLUTO software (Karypis, 2003). Two biomedical data sets from the Kent Ridge Biomedical Data Set Repository (Li & Liu, 2003) were also used in our experiments. Besides the above high-dimensional data sets, we also used two UCI data sets (Newman, Hettich, Blake, & Merz, 1998) with normal dimensionality.

The Uniform Effect of K-means

Table 1 shows the experimental results by K-means. As can be seen, for 15 out of 16 data sets, K-means reduced the variation of the cluster sizes, as indicated by the negative DCV values. We can also capture this information from Figure 5. That is, the square-line figure representing the CV_1 values by K-means is almost enveloped by the dot-line figure, i.e., the CV_0 values of all data sets. This result indicates that, for data sets with high variation on the "true" cluster sizes, the uniform effect of K-means is dominant. Furthermore, an interesting observation is that, while the range of CV_0 is between 0.27 and 1.95, the range of CV_1 by K-means is restricted into a much smaller range from 0.33 to 0.94. Thus the empirical interval of CV_1 values is [0.3, 1.0].

Table 1. Data sets and the clustering results

Data set	#objects	#features	#classes	CV_0	K-means		UPGMA	
					CV_1	DCV	CV_1	DCV
Document Data Set								
fbis	2463	2000	17	0.961	0.551	-0.410	1.491	0.530
hitech	2301	126373	6	0.495	0.365	-0.130	2.410	1.915
k1a	2340	21839	20	1.004	0.490	-0.514	1.714	0.710
k1b	2340	21839	6	1.316	0.652	-0.664	1.647	0.331
la2	3075	31472	6	0.516	0.377	-0.139	1.808	1.292
la12	6279	31472	6	0.503	0.461	-0.042	2.437	1.934
ohscal	11162	11465	10	0.266	0.438	0.172	1.634	1.368
re1	1657	3758	25	1.385	0.325	-1.060	1.830	0.445
re0	1504	2886	13	1.502	0.390	-1.112	2.264	0.762
tr31	927	10128	7	0.936	0.365	-0.571	1.152	0.216
tr41	878	7454	10	0.913	0.393	-0.520	0.984	0.071
wap	1560	8460	20	1.040	0.494	-0.547	1.612	0.572
Biomedical Data Set								
Leukemia	325	12558	7	0.584	0.369	-0.212	1.770	1.186
LungCancer	203	12600	5	1.363	0.631	-0.731	1.888	0.525
UCI Data Set								
ecoli	336	7	8	1.160	0.498	-0.662	1.804	0.644
page-blocks	5473	10	5	1.953	0.940	-1.013	2.023	0.070
Min	203	7	5	0.266	0.325	-1.112	0.984	0.070
Max	11162	126373	25	1.953	0.940	0.172	2.437	1.934

Figure 5. A comparison of CV_1 values by k-means and UPGMA on all data sets

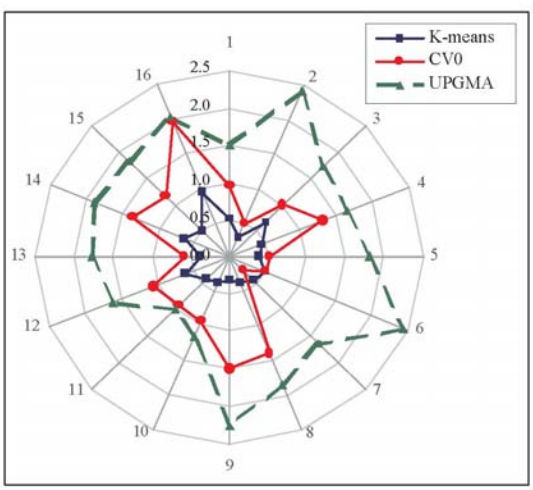
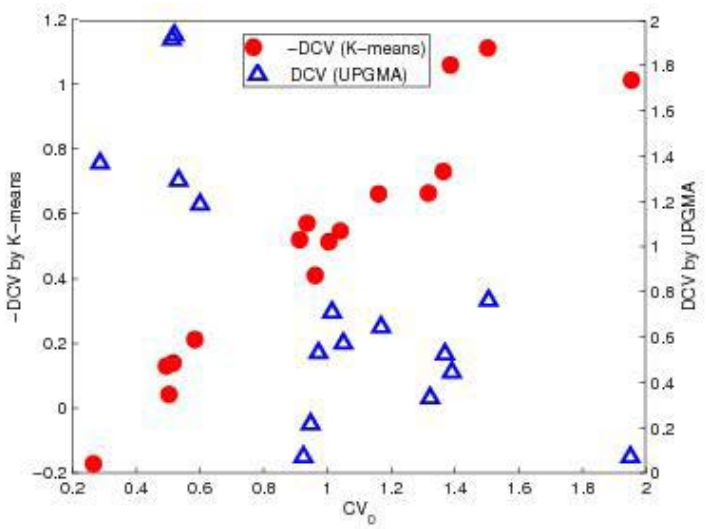


Figure 6. The problems of the two effects



The Dispersion Effect of UPGMA

Table 1 also shows the experimental results by UPGMA. As can be seen, for all data sets, UPGMA tends to increase the variation of the cluster sizes. Indeed, no matter what the CV_0 values are, the corresponding CV_1 values are no less than 1.0, with only one exception: the "tr41" data set. Therefore, we can empirically estimate the interval of CV_1 values as [1.0, 2.5]. Also, the radar figure in Figure 5 directly illustrates the dispersion effect of UPGMA, in which the triangle-dashed line figure of UPGMA envelops the other two figures.

Discussions on the Two Diverse Effects

From the above illustrations and experimental validations, we can draw the conclusion that the uniform effect of K-means and the dispersion effect of UPGMA hold for many real-world data sets. This implies that the distribution of the "true" cluster sizes can make impact on the performance of the clustering algorithms.

Consider the data set "re0" in table 1 which consists of highly imbalanced "true" clusters, i.e., CV_0= 1.502. If K-means is employed on this data set, the resultant cluster sizes can be much more uniform, say CV_1=0.390, due to the uniform effect of K-means. The great difference on the CV values indicates that the clustering result is very poor indeed. In fact, as Figure 6 shows, the absolute DCV values increase linearly as the CV_0 values increase when employing K-means clustering. Therefore, for data sets with high variation on the "true" cluster sizes, i.e., CV_0>>1.0, K-means is not a good choice for the clustering task.

Consider another data set "ohscal" in table 1 which contains rather uniform "true" clusters, i.e., CV_0=0.266. After employing UPGMA, however, the CV_1 value of the resultant cluster sizes goes up to 1.634, which also indicates a very poor clustering quality. Actually, as Figure 6 shows, due to the dispersion effect of UPGMA, the DCV values increase rapidly as the CV_0 values decrease. Therefore, for data sets with low variation on the "true" cluster sizes, i.e., CV_0<<1.0, UPGMA is not an appropriate clustering algorithm.

FUTURE TRENDS

As we know, data mining emerges as a prosperous discipline when the real-world practices produce more and more new data sets. To analyze these data sets is rather challenging, since the traditional data analysis techniques may encounter difficulties (such as addressing the problems of high dimensionality and scalability), or many intrinsic data factors can not be well characterized or even be totally unknown to us. The hardness of solving the former problem is obvious, but the latter problem is more "dangerous" – we may get a poor data analysis result without knowing this due to the data characteristics inherently. For example, applying K-means on highly imbalanced data set is often ineffective, and the ineffectiveness may not be identified by some cluster validity measures (Xiong, Wu, & Chen, 2006), thus the clustering result can be misleading. Also, the density of data in the feature space can be uneven, which can bring troubles to the clustering algorithms based on the graph partitioning scheme. Moreover, it is well-known that outliers and noise in the data can disturb the clustering process; however, how to identify them is still an open topic in the field of anomaly detection. So in the future, we can expect that, more research efforts will be put on capturing and characterizing the intrinsic data factors (known or unknown) and their impacts on the data analysis results. This is crucial for the successful utilization of data mining algorithms in practice.

CONCLUSION

In this chapter, we illustrate the relationship between the clustering algorithms and the distribution of the "true" cluster sizes. Our experimental results show that K-means tends to reduce the variation on the cluster sizes if the "true" cluster sizes are highly imbalanced, and UPGMA tends to increase the variation on the cluster sizes if the "true" cluster sizes are relatively uniform. Indeed, the variations of the resultant cluster sizes by K-means and UPGMA, measured by the Coefficient of Variation (CV), are in the specific intervals, say [0.3, 1.0] and [1.0, 2.5] respectively. Therefore, for data sets with highly imbalanced "true" cluster sizes, i.e., CV_0>>1.0, K-means may not be the good choice of clustering algorithms. However, for data sets with low variation on the "true" cluster sizes, i.e., CV_0<<1.0, UPGMA may not be an appropriate clustering algorithm.

REFERENCES

Bradley, P., Fayyad, U., & Reina, C. (1998). Scaling clustering algorithms to large databases. In *Proc. of the 4th ACM SIGKDD*, 9-15.

Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. (2000). Lof: Identifying density based local outliers. In *Proc. of the 2000 ACM SIGMOD*, 93-104.

DeGroot, M., & Schervish, M. (2001). *Probability and Statistics*, 3rd Ed. Addison Wesley.

Ertoz, L., Steinbach, M., & Kumar, V. (2002). A new shared nearest neighbor clustering algorithm and its applications. In *Proc. of Workshop on Clustering High Dimensional Data and its Applications, the 2nd SIAM SDM*, 105-115.

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. of the 2th ACM SIGKDD*, 226–231.

Ghosh, J. (2003). *Scalable Clustering Methods for Data Mining, Handbook of Data Mining*. Lawrence Ealbaum Assoc.

Guha, S., Rastogi, R., & Shim, K. (1998). Cure: An efficient clustering algorithm for large databases. In *Proc. of the 1998 ACM SIGMOD*, 73–84.

Jain, A. K., & Dubes, R. C. (1988). *Algorithms for Clustering Data*. Prentice Hall.

Jarvis, R. A., & Patrick, E. A. (1973). Clustering using a similarity measure based on shared nearest neighbors. *IEEE Transactions on Computers*, C-22, 1025–1034.

Karypis, G. (2003). *CLUTO – Software for Clustering High-dimensional Datasets*, version 2.1.1., from http://glaros.dtc.umn.edu/gkhome/views/cluto.

Karypis, G., Han, E.-H., & Kumar, V. (1999). Chameleon: A hierarchical clustering algorithm using dynamic modeling. *IEEE Computer*, 32(8), 68–75.

Kent, J. T., Bibby, J. M., & Mardia, K. V. (2006). *Multivariate Analysis* (*Probability and Mathematical Statistics*). Elsevier Limited.

Li, J., & Liu, H. (2003). *Kent Ridge Biomedical Data Set Repository*, from http://sdmc.i2r.a-star.edu.sg/rp/.

Murtagh, F. (2000). *Clustering Massive Data Sets*, *Handbook of Massive Data Sets*. Kluwer.

Newman, D., Hettich, S., Blake, C., & Merz, C. (1998). *UCI Repository of Machine Learning Databases*, from http://www.ics.uci.edu/~mlearn/MLRepository.html.

Ng, R. T., & Han., J. (2002). Clarans: A method for clustering objects for spatial data mining. *IEEE Transactions on Knowledge and Data Engineering*, 14(5), 1003–1016.

Tan, P.-N., Steinbach, M., & Kumar, V. (2005). *Introduction to Data Mining*. Addison-Wesley.

Tang, Bin, Shepherd, M., Heywood, M. I., & Luo, X. (2005). Comparing dimension reduction techniques for document clustering. In *Proc. of Canadian Conference on AI*, 292–296.

Wu, J., Xiong, H., Wu, P., & Chen, J. (2007). Local decomposition for rare class analysis. In *Proc. of the 13th ACM SIGKDD*, in press.

Xiong, H., Pandey, G., Steinbach, M., & Kumar, V. (2006). Enhancing data analysis with noise removal. *IEEE Transactions on Knowledge and Data Engineering*, 18(3), 304–319.

Xiong, H., Wu, J., & Chen, J. (2006). K-means clustering versus validation measures: a data distribution perspective. In *Proc. of the 12th ACM SIGKDD*, 779-784.

Zhang, T., Ramakrishnan, R., & Livny, M. (1996). Birch: An efficient data clustering method for very large databases. In *Proc. of the 1996 ACM SIGMOD*, 103–114.

Zhao, Y., & Karypis, G. (2004). Criterion functions for document clustering: Experiments and analysis. *Machine Learning*, 55(3), 311–331.

KEY TERMS

Cluster Analysis: Cluster analysis provides insight into the data by dividing the objects into clusters of objects, such that objects in a cluster are more similar to each other than objects in other clusters. In pattern recognition community, it is often called "unsupervised learning".

Coefficient of Variation (CV): A statistic that measures the dispersion of a data vector. CV is defined as the ratio of the standard deviation to the mean.

Dispersion Effect: If the distribution of the cluster sizes by a clustering method has more variation than the distribution of the "true" cluster sizes, we say that this clustering method shows the dispersion effect.

Hierarchical Clustering: Hierarchical clustering analysis provides insight into the data by assembling all the objects into a dendrogram, such that each sub-cluster is a node of the dendrogram, and the combinations of sub-clusters create a hierarchy. The process of the hierarchical clustering can be either divisive or agglomerative. Single-link, complete-link and UPGMA are the well-known agglomerative hierarchical clustering methods.

High Dimensional Data: The data with hundreds or thousands of dimensions from the areas of text/web mining, bioinformatics, etc. Also, these data sets are often sparse. Analyzing such high-dimensional data sets is a contemporary challenge.

K-Means: A prototype-based, simple partitional clustering technique which attempts to find a user-specified *K* number of clusters.

Uniform Effect: If the distribution of the cluster sizes by a clustering method has less variation than the distribution of the "true" cluster sizes, we say that this clustering method shows the uniform effect.

Data Driven vs. Metric Driven Data Warehouse Design

John M. Artz
The George Washington University, USA

INTRODUCTION

Although data warehousing theory and technology have been around for well over a decade, they may well be the next hot technologies. How can it be that a technology sleeps for so long and then begins to move rapidly to the foreground? This question can have several answers. Perhaps the technology had not yet caught up to the theory or that computer technology 10 years ago did not have the capacity to delivery what the theory promised. Perhaps the ideas and the products were just ahead of their time. All these answers are true to some extent. But the real answer, I believe, is that data warehousing is in the process of undergoing a radical theoretical and paradigmatic shift, and that shift will reposition data warehousing to meet future demands.

BACKGROUND

Just recently I started teaching a new course in data warehousing. I have only taught it a few times so far, but I have already noticed that there are two distinct and largely incompatible views of the nature of a data warehouse. A prospective student, who had several years of industry experience in data warehousing but little theoretical insight, came by my office one day to find out more about the course. "Are you an Inmonite or a Kimballite?" she inquired, reducing the possibilities to the core issues. "Well, I suppose if you put it that way," I replied, "I would have to classify myself as a Kimballite." William Inmon (2000, 2002) and Ralph Kimball (1996, 1998, 2000) are the two most widely recognized authors in data warehousing and represent two competing positions on the nature of a data warehouse.

The issue that this student was trying to get at was whether or not I viewed the dimensional data model as the core concept in data warehousing. I do, of course, but there is, I believe, a lot more to the emerging competition between these alternative views of data warehouse design. One of these views, which I call the data-driven view of data warehouse design, begins with existing organizational data. These data have more than likely been produced by existing transaction processing systems. They are cleansed and summarized and are used to gain greater insight into the functioning of the organization. The analysis that can be done is a function of the data that were collected in the transaction processing systems. This was, perhaps, the original view of data warehousing and, as will be shown, much of the current research in data warehousing assumes this view.

The competing view, which I call the metric-driven view of data warehouse design, begins by identifying key business processes that need to be measured and tracked over time in order for the organization to function more efficiently. A dimensional model is designed to facilitate that measurement over time, and data are collected to populate that dimensional model. If existing organizational data can be used to populate that dimensional model, so much the better. But if not, the data need to be acquired somehow. The metric-driven view of data warehouse design, as will be shown, is superior both theoretically and philosophically. In addition, it dramatically changes the research program in data warehousing. The metric-driven and data-driven approaches to data warehouse design have also been referred to, respectively, as metric pull versus data push (Artz, 2003).

MAIN THRUST

Data-Driven Design

The classic view of data warehousing sees the data warehouse as an extension of decision support systems. Again, in a classic view, decision support systems sit atop management information systems and use data extracted from management information and transaction processing systems to support decisions within

the organization. This view can be thought of as a data-driven view of data warehousing, because the exploitations that can be done in the data warehouse are driven by the data made available in the underlying operational information systems.

This data-driven model has several advantages. First, it is much more concrete. The data in the data warehouse are defined as an extension of existing data. Second, it is evolutionary. The data warehouse can be populated and exploited as new uses are found for existing data. Finally, there is no question that summary data can be derived, because the summaries are based upon existing data. However, it is not without flaws. First, the integration of multiple data sources may be difficult. These operational data sources may have been developed independently, and the semantics may not agree. It is difficult to resolve these conflicting semantics without a known end state to aim for. But the more damaging problem is epistemological. The summary data derived from the operational systems represent something, but the exact nature of that something may not be clear. Consequently, the meaning of the information that describes that something may also be unclear. This is related to the semantic disintegrity problem in relational databases. A user asks a question of the database and gets an answer, but it is not the answer to the question that the user asked. When the somethings that are represented in the database are not fully understood, then answers derived from the data warehouse are likely to be applied incorrectly to known somethings. Unfortunately, this also undermines data mining. Data mining helps people find hidden relationships in the data. But if the data do not represent something of interest in the world, then those relationships do not represent anything interesting, either.

Research problems in data warehousing currently reflect this data-driven view. Current research in data warehousing focuses on a) data extraction and integration, b) data aggregation and production of summary sets, c) query optimization, and d) update propagation (Jarke, Lenzerini, Vassiliou, & Vassiliadis, 2000). All these issues address the production of summary data based on operational data stores.

A Poverty of Epistemology

The primary flaw in data-driven data warehouse design is that it is based on an impoverish epistemology. *Epistemology* is that branch of philosophy concerned with theories of knowledge and the criteria for valid knowledge (Fetzer & Almeder, 1993; Palmer, 2001). That is to say, when you derive information from a data warehouse based on the data-driven approach, what does that information mean? How does it relate to the work of the organization? To see this issue, consider the following example. If I asked each student in a class of 30 for their ages, then summed those ages and divided by 30, I should have the average age of the class, assuming that everyone reported their age accurately. If I were to generate a list of 30 random numbers between 20 and 40 and took the average, that average would be the average of the numbers in that data set and would have nothing to do with the average age of the class. In between those two extremes are any number of options. I could guess the ages of students based on their looks. I could ask members of the class to guess the ages of other members. I could rank the students by age and then use the ranking number instead of age. The point is that each of these attempts is somewhere between the two extremes, and the validity of my data improves as I move closer to the first extreme. That is, I have measurements of a specific phenomenon, and those measurements are likely to represent that phenomenon faithfully. The epistemological problem in data-driven data warehouse design is that data is collected for one purpose and then used for another purpose. The strongest validity claim that can be made is that any information derived from this data is true about the data set, but its connection to the organization is tenuous. This not only creates problems with the data warehouse, but all subsequent data-mining discoveries are suspect also.

METRIC-DRIVEN DESIGN

The metric-driven approach to data warehouse design begins by defining key business processes that need to be measured and tracked in order to maintain or improve the efficiency and productivity of the organization. After these key business processes are defined, they are modeled in a dimensional data model and then further analysis is done to determine how the dimensional model will be populated. Hopefully, much of the data can be derived from operational data stores, but the metrics are the driver, not the availability of data from operational data stores.

A relational database models the entities or objects of interest to an organization (Teorey, 1999). These objects of interest may include customers, products, employees, and the like. The entity model represents these things and the relationships between them. As occurrences of these entities enter or leave the organization, that addition or deletion is reflected in the database. As these entities change in state, somehow, those state changes are also reflected in the database. So, theoretically, at any point in time, the database faithfully represents the state of the organization. Queries can be submitted to the database, and the answers to those queries should, indeed, be the answers to those questions if they were asked and answered with respect to the organization.

A data warehouse, on the other hand, models the business processes in an organization to measure and track those processes over time. Processes may include sales, productivity, the effectiveness of promotions, and the like. The dimensional model contains facts that represent measurements over time of a key business process. It also contains dimensions that are attributes of these facts. The fact table can be thought of as the dependent variable in a statistical model, and the dimensions can be thought of as the independent variables. So the data warehouse becomes a longitudinal dataset tracking key business processes.

A Parallel with Pre-Relational Days

You can see certain parallels between the state of data warehousing and the state of database prior to the relational model. The relational model was introduced in 1970 by Codd but was not realized in a commercial product until the early 1980s (Date, 2004). At that time, a large number of nonrelational database management systems existed. All these products handled data in different ways, because they were software products developed to handle the problem of storing and retrieving data. They were not developed as implementations of a theoretical model of data. When the first relational product came out, the world of databases changed almost overnight. Every nonrelational product attempted, unsuccessfully, to claim that it was really a relational product (Codd, 1985). But no one believed the claims, and the nonrelational products lost their market share almost immediately.

Similarly, a wide variety of data warehousing products are on the market today. Some are based on the dimensional model, and some are not. The dimensional model provides a basis for an underlying theory of data that tracks processes over time rather than the current state of entities. Admittedly, this model of data needs quite a bit of work, but the relational model did not come into dominance until it was coupled with entity theory, so the parallel still holds. We may never have an announcement in data warehousing as dramatic as Codd's paper in relational theory. It is more likely that a theory of temporal dimensional data will accumulate over time. However, in order for data warehousing to become a major force in the world of databases, an underlying theory of data is needed and will eventually be developed.

The Implications for Research

The implications for research in data warehousing are rather profound. Current research focuses on issues such as data extraction and integration, data aggregation and summary sets, and query optimization and update propagation. All these problems are applied problems in software development and do not advance our understanding of the theory of data.

But a metric-driven approach to data warehouse design introduces some problems whose resolution can make a lasting contribution to data theory. Research problems in a metric-driven data warehouse include a) How do we identify key business processes? b) How do we construct appropriate measures for these processes? c) How do we know those measures are valid? d) How do we know that a dimensional model has accurately captured the independent variables? e) Can we develop an abstract theory of aggregation so that the data aggregation problem can be understood and advanced theoretically? and, finally, f) can we develop an abstract data language so that aggregations can be expressed mathematically by the user and realized by the machine?

Initially, both data-driven and metric-driven designs appear to be legitimate competing paradigms for data warehousing. The epistemological flaw in the data-driven approach is a little difficult to grasp, and the distinction — that information derived from a data-driven model is information about the data set, but information derived from a metric-driven model is information about the organization — may also be a bit elusive. However, the implications are enormous. The data-driven model has little future in that it is founded on a model of data exploitation rather than a model of

data. The metric-driven model, on the other hand, is likely to have some major impacts and implications.

FUTURE TRENDS

The Impact on White-Collar Work

The data-driven view of data warehousing limits the future of data warehousing to the possibilities inherent in summarizing large collections of old data without a specific purpose in mind. The metric-driven view of data warehousing opens up vast new possibilities for improving the efficiency and productivity of an organization by tracking the performance of key business processes. The introduction of quality management procedures in manufacturing a few decades ago dramatically improved the efficiency and productivity of manufacturing processes, but such improvements have not occurred in white-collar work.

The reason that we have not seen such an improvement in white-collar work is that we have not had metrics to track the productivity of white-collar workers. And even if we did have the metrics, we did not have a reasonable way to collect them and track them over time. The identification of measurable key business processes and the modeling of those processes in a data warehouse provides the opportunity to perform quality management and process improvement on white-collar work.

Subjecting white-collar work to the same rigorous definition as blue-collar work may seem daunting, and indeed that level of definition and specification will not come easily. So what would motivate a business to do this? The answer is simple: Businesses will have to do this when the competitors in their industry do it. Whoever does this first will achieve such productivity gains that competitors will have to follow suit in order to compete. In the early 1970s, corporations were not revamping their internal procedures because computerized accounting systems were fun. They were revamping their internal procedures because they could not protect themselves from their competitors without the information for decision making and organizational control provided by their accounting information systems. A similar phenomenon is likely to drive data warehousing.

Dimensional Algebras

The relational model introduced *Structured Query Language (SQL),* an entirely new data language that allowed nontechnical people to access data in a database. SQL also provided a means of thinking about record selection and limited aggregation. Dimensional models can be exploited by a dimensional query language such as MDX (Spofford, 2001), but much greater advances are possible.

Research in data warehousing will likely yield some sort of a dimensional algebra that will provide, at the same time, a mathematical means of describing data aggregation and correlation and a set of concepts for thinking about aggregation and correlation. To see how this could happen, think about how the relational model led us to think about the organization as a collection of entity types or how statistical software made the concepts of correlation and regression much more concrete.

A Unified Theory Of Data

In the organization today, the database administrator and the statistician seem worlds apart. Of course, the statistician may have to extract some data from a relational database in order to do his or her analysis. And the statistician may engage in limited data modeling in designing a data set for analysis by using a statistical tool. The database administrator, on the other hand, will spend most of his or her time in designing, populating, and maintaining a database. A limited amount of time may be devoted to statistical thinking when counts, sums, or averages are derived from the database. But these two individuals will largely view themselves as participating in greatly differing disciplines.

With dimensional modeling, the gap between database theory and statistics begins to close. In dimensional modeling we have to begin thinking in terms of construct validity and temporal data. We need to think about correlations between dependent and independent variables. We begin to realize that the choice of data types (e.g., interval or ratio) will affect the types of analysis we can do on the data and will hence potentially limit the queries. So the database designer has to address concerns that have traditionally been the domain of the statistician. Similarly, the statistician cannot afford the luxury of constructing a data set for a single purpose or a single type of analysis. The data

set must be rich enough to allow the statistician to find relationships that may not have been considered when the data set was being constructed. Variables must be included that may potentially have impact, may have impact at some times but not others, or may have impact in conjunction with other variables. So the statistician has to address concerns that have traditionally been the domain of the database designer.

What this points to is the fact that database design and statistical exploitation are just different ends of the same problem. After these two ends have been connected by data warehouse technology, a single theory of data must be developed to address the entire problem. This unified theory of data would include entity theory and measurement theory at one end and statistical exploitation at the other. The middle ground of this theory will show how decisions made in database design will affect the potential exploitations, so intelligent design decisions can be made that will allow full exploitation of the data to serve the organization's needs to model itself in data.

CONCLUSION

Data warehousing is undergoing a theoretical shift from a data-driven model to a metric-driven model. The metric-driven model rests on a much firmer epistemological foundation and promises a much richer and more productive future for data warehousing. It is easy to haze over the differences or significance between these two approaches today. The purpose of this article was to show the potentially dramatic, if somewhat speculative, implications of the metric-driven approach.

REFERENCES

Artz, J. (2003). Data push versus metric pull: Competing paradigms for data warehouse design and their implications. In M. Khosrow-Pour (Ed.), *Information technology and organizations: Trends, issues, challenges and solutions*. Hershey, PA: Idea Group Publishing.

Codd, E. F. (1970). A relational model of data for large shared data banks. *Communications of the ACM, 13*(6), 377-387.

Codd, E. F. (1985, October 14). Is your DBMS really relational? *Computerworld*.

Date, C. J. (2004). *An introduction to database systems* (8th ed.). Addison-Wesley.

Fetzer, J., & Almeder, F. (1993). *Glossary of epistemology/philosophy of science.* Paragon House.

Inmon, W. (2002). *Building the data warehouse.* Wiley.

Inmon, W. (2000). *Exploration warehousing: Turning business information into business opportunity.* Wiley.

Jarke, M., Lenzerini, M., Vassiliou, Y., & Vassiliadis, P. (2000). *Fundamentals of data warehouses.* Springer-Verlag.

Kimball, R. (1996). *The data warehouse toolkit.* Wiley.

Kimball, R. (1998). *The data warehouse lifecycle toolkit.* Wiley.

Kimball, R., & Merz, R. (2000). *The data webhouse toolkit.* Wiley.

Palmer, D. (2001). *Does the center hold? An introduction to Western philosophy.* McGraw-Hill.

Spofford, G. (2001). *MDX Solutions.* Wiley.

Teorey, T. (1999). *Database modeling & design.* Morgan Kaufmann.

Thomsen, E. (2002). *OLAP solutions.* Wiley.

KEY TERMS

Data-Driven Design: A data warehouse design that begins with existing historical data and attempts to derive useful information regarding trends in the organization.

Data Warehouse: A repository of time-oriented organizational data used to track the performance of key business processes.

Dimensional Data Model: A data model that represents measurements of a process and the independent variables that may affect that process.

Epistemology: A branch of philosophy that attempts to determine the criteria for valid knowledge.

Key Business Process: A business process that can be clearly defined, is measurable, and is worthy of improvement.

Metric-Driven Design: A data warehouse design that beings with the definition of metrics that can be used to track key business processes.

Relational Data Model: A data model that represents entities in the form of data tables.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 223-227, copyright 2005 by IGI Publishing, formerly known as Idea Group Publishing (an imprint of IGI Global).

Data Mining and Privacy

Esma Aïmeur
Université de Montréal, Canada

Sébastien Gambs
Université de Montréal, Canada

INTRODUCTION

With the emergence of Internet, it is now possible to connect and access sources of information and databases throughout the world. At the same time, this raises many questions regarding the privacy and the security of the data, in particular how to mine useful information while preserving the privacy of sensible and confidential data. *Privacy-preserving data mining* is a relatively new but rapidly growing field that studies how data mining algorithms affect the privacy of data and tries to find and analyze new algorithms that preserve this privacy.

At first glance, it may seem that data mining and privacy have orthogonal goals, the first one being concerned with the discovery of useful knowledge from data whereas the second is concerned with the protection of data's privacy. Historically, the interactions between privacy and data mining have been questioned and studied since more than a decade ago, but the name of the domain itself was coined more recently by two seminal papers attacking the subject from two very different perspectives (Agrawal & Srikant, 2000; Lindell & Pinkas, 2000). The first paper (Agrawal & Srikant, 2000) takes the approach of randomizing the data through the injection of noise, and then recovers from it by applying a reconstruction algorithm before a learning task (the induction of a decision tree) is carried out on the reconstructed dataset. The second paper (Lindell & Pinkas, 2000) adopts a cryptographic view of the problem and rephrases it within the general framework of secure multiparty computation.

The outline of this chapter is the following. First, the area of privacy-preserving data mining is illustrated through three scenarios, before a classification of privacy-preserving algorithms is described and the three main approaches currently used are detailed. Finally, the future trends and challenges that await the domain are discussed before concluding.

BACKGROUND

The area of privacy-preserving data mining can still be considered in its infancy but there are already several workshops (usually held in collaboration with different data mining and machine learning conferences), two different surveys (Verykios *et al.*, 2004; Výborný, 2006) and a short book (Vaidya, Clifton & Zhu, 2006) on the subject. The notion of privacy itself is difficult to formalize and quantify, and it can take different flavours depending on the context. The three following scenarios illustrate how privacy issues can appear in different data mining contexts.

- **Scenario 1:** A famous Internet-access provider wants to release the log data of some of its customers (which include their personal queries over the last few months) to provide a public benchmark available to the web mining community. How can the company anonymize the database in such a way that it can guarantee to its clients that no important and sensible information can be mined about them?
- **Scenario 2:** Different governmental agencies (for instance the Revenue Agency, the Immigration Office and the Ministry of Justice) want to compute and release some joint statistics on the entire population but they are constrained by the law not to communicate any individual information on citizens, even to other governmental agencies. How can the agencies compute statistics that are sufficiently accurate while at the same time, safeguarding the privacy of individual citizens?

- **Scenario 3:** Consider two bioinformatics companies: Alice Corporation and Bob Trust. Each company possesses a huge database of bioinformatics data gathered from experiments performed in their respective labs. Both companies are willing to cooperate in order to achieve a learning task of mutual interest such as a clustering algorithm or the derivation of association rules, nonetheless they do not wish to exchange their whole databases because of obvious privacy concerns. How can they achieve this goal without disclosing any unnecessary information?

When evaluating the potential privacy leak caused by a data mining process, it is important to keep in mind that the adversary may have some side information that could be used to infringe this privacy. Indeed, while the data mining process by itself may not be directly harmful, it is conceivable that associated with the help of linking attacks (derived from some *a priori* knowledge), it may lead to a total breakdown of the privacy.

MAIN FOCUS

The privacy-preserving techniques can generally be classified according to the following dimensions:

- *The distribution of the data*. During the data mining process, the data can be either in the hands of a single entity or distributed among several participants. In the case of distributed scenarios, a further distinction can be made between the situation where the attributes of a single record are split among different sites (*vertical partitioning*) and the case where several databases are situated in different locations (*horizontal partitioning*). For example, in scenario 1 all the data belongs to the Internet provider, whereas in scenario 2 corresponds to a vertical partitioning of the data where the information on a single citizen is split among the different governmental agencies and scenario 3 corresponds to an horizontal partitioning.
- *The data mining algorithm*. There is not yet a single generic technique that could be applied to any data mining algorithm, thus it is important to decide beforehand which algorithm we are interested in. For instance, privacy-preserving variants of association rules, decision trees, neural networks, support vector machines, boosting and clustering have been developed.
- *The privacy-preservation technique*. Three main families of privacy-preserving techniques exist: the *perturbation-based approaches*, the *randomization methods* and the *secure multiparty solutions*. The first two families protect the privacy of data by introducing noise whereas the last family uses cryptographic tools to achieve privacy-preservation. Each technique has his pros and cons and may be relevant in different contexts. The following sections describe and explicit these three privacy-preservation techniques.

PERTURBATION-BASED APPROACHES

The perturbation-based approaches rely on the idea of *modifying the values of selected attributes using heuristics in order to protect the privacy of the data*. These methods are particularly relevant when the dataset has to be altered so that it preserves privacy before it can be released publicly (such as in scenario 1 for instance). Modifications of the data can include:

- *Altering the value of a specific attribute* by either perturbing it (Atallah *et al.*, 1999) or replacing it by the "unknown" value (Chang & Moskowitz, 2000).
- *Swapping the value of an attribute* between two individual records (Fienberg & McIntyre, 2004).
- *Using a coarser granularity* by merging several possible values of an attribute into a single one (Chang & Moskowitz, 2000).

This process of increasing uncertainty in the data in order to preserve privacy is called *sanitization*. Of course, introducing noise in the data also decreases the utility of the dataset and renders the learning task more difficult. Therefore, there is often a compromise to be made between the privacy of the data and how useful is the sanitized dataset. Moreover, finding the optimal way to sanitize the data has been proved to be a NP-hard problem in some situations (Meyerson & Williams, 2004). However, some sanitization procedures offer privacy guarantees about how hard it is to pinpoint a particular individual. For instance, the

k-anonymization procedure (Sweeney, 2002), which proceeds through suppression and generalizations of attribute values, generates a *k*-anonymized dataset in which each individual record is indistinguishable from at least *k*-1 other records within the dataset. This guarantees that no individual can be pinpointed with probability higher than 1/*k*-1, even with the help of linking attacks.

RANDOMIZATION METHODS

While the perturbation-based approaches will alter the value of individual records, the randomization methods *act on the data in a global manner by adding independent Gaussian or uniform noise to the attribute values*. The goal remains, as for the perturbation approach, to hide particular attribute values while preserving the joint distribution of the data. The "polluted" data set can either be used to reconstruct the data distribution, for instance by using an Expectation-Maximization type of algorithm (Agrawal & Aggarwal, 2001), or to build directly a classifier out of the noisy data (Agrawal & Srikant, 2000). The randomization methods are well adapted to the situation where the data is distributed (horizontally and/or vertically) and the participants are willing to send a randomized version of their data to a central site which will be responsible to perform the data mining process (see for example the related work on privacy-preserving online analytical processing (Agrawal, Srikant & Dilys, 2005)). For instance, in scenario 2 the different governmental agencies could agree to send a randomized version of their dataset to a *semi-trusted* third party that will carry out the computation of the data mining algorithm and then release publicly the results.

The intensity and the type of the noise are the parameters that can be tuned to balance between the data privacy and the accuracy of the model. Two notions are commonly used to measure privacy: the *conditional entropy* (Agrawal & Aggarwal, 2001) and *privacy breaches* (Evfimievski, Gerhke & Srikant, 2003). Conditional entropy is an information-theoretic measure that computes the mutual information between the original and the randomized dataset. Low mutual information is generally an indication of high privacy preservation but also a sign that the learning will be less accurate. A privacy breach might occur when there is a change of confidence regarding the estimated value of a particular attribute in a specific record. Evfimievski, Gerhke and Srikant (2003) were the first to formally analyze the trade-off between noise addition and data utility and to propose a method which limits privacy breaches without any knowledge of the data distribution.

SECURE MULTIPARTY SOLUTIONS

Secure multiparty computation is the branch of cryptography which deals with the realization of distributed tasks in a secure manner (Goldreich, 2004). The definition of security usually includes preserving the privacy of the data and protecting the computation against malicious participants. A typical task is to compute some function $f(x,y)$ in which the input x is in the hands of one participant while input y is the hands of the other. An example of possible inputs could be the personal datasets of two companies (for instance Alice Corporation and Bob Trust in scenario 3) and the output of the function could be for instance a classifier or a set of rules jointly extracted from their datasets. Within the usual cryptographic paradigm, a protocol is considered perfectly secure if its execution does not reveal more information than the protocol itself. In particular, the participants should not be able to learn anything from the completion of the task, except what they can infer from their own inputs and the output of the function (such as the resulting classifier).

Two security models are generally considered in cryptography (Goldreich, 2004). In the *honest-but-curious* (also known as passive or semi-honest) model, the participants are assumed to follow the directives of the protocol, without any attempt to cheat, but they may try to learn as much information as possible by analyzing the communication exchanged during the protocol. This model is weaker than if we had considered the model where the participants can be *malicious* and cheat *actively* during the execution of the protocol, but it is nonetheless useful and almost always considered by privacy-preserving algorithms. In particular, the semi-honest model is relevant in the situation where the participants are willing to cooperate to achieve a task of mutual interest but cannot communicate directly their dataset because of confidentiality issues.

General results in the domain guarantee that any function can be implemented with unconditional security (in the sense of information theory) provided that at least some proportion of the participants is honest

(see for instance Chaum, Crépeau & Damgård, 1988). Although these methods are very generic and universal, they can be quite inefficient in terms of communication and computational complexity, when the function to compute is complex and the input data is large (which is typically the case in data-mining). Therefore, it is usually worthwhile to develop an efficient secure implementation of a specific learning task that offers a lower communicational and computational cost than the general technique. Historically, Lindell and Pinkas (2000) were the first to propose an algorithm that followed this approach. They described a privacy-preserving, distributed bipartite version of ID3. Since this seminal paper, other efficient privacy-preserving variants of learning algorithms have been developed such as neural networks (Chang & Lu, 2001), naïve Bayes classifier (Kantarcioglŭ & Vaidya, 2004), *k*-means (Kruger, Jha & McDaniel, 2005), support vector machines (Laur, Lipmaa & Mielikäinen, 2006) and boosting (Gambs, Kégl & Aïmeur, 2007).

FUTURE TRENDS

Understanding privacy better and finding relevant measures to compare different privacy-preserving approaches are some of the major challenges in the domain. For instance, it seems that the traditional "model" of secure multiparty computation may not be perfectly adequate for data mining. In particular, the participants of a multiparty protocol might not be only interested in how much information is leaked during the computation of the function *f*, but rather in the total direct information on their datasets that is revealed, including the information contained in *f* itself. At least two directions of research could be explored to solve this problem. First, some studies have already considered the computation of an *approximation* of the function *f* instead of the function itself (Feigenbaum *et al.*, 2001). Approximation algorithms are usually utilized when the exact version of the function is hard to compute. Nonetheless, approximation algorithms could also be used with a different purpose in mind, such as hiding information about the function (see for instance Ishai *et al.*, 2007). The second direction to explore is the introduction of some kind of randomization in the learning process (and not in the data itself). Some algorithms, such as neural networks with randomly initialized weights, seem naturally suited for this type of approach, while others could be changed explicitly to include randomization.

Another important research avenue is the development of *hybrid algorithms* that are able to take the best of both worlds. For instance, the randomization approach could be used as preprocessing to disturb the data before a secure multiparty protocol is run on it. This guarantees that even if the cryptographic protocol is broken, some privacy of the original dataset will still be preserved.

Finally, a last trend of research concerns the migration from the semi-honest setting, where participants are considered to be honest-but-curious, to the malicious setting where participants may actively cheat and try to tamper with the output of the protocol. Moreover, it is important to consider the situation where a participant may be involved in the data mining process but without putting his own data at stake and instead run the algorithm on a deliberately forged dataset. Is there a way to detect and avoid this situation? For instance by forcing the participant to commit his dataset (in a cryptographic sense) before the actual protocol is run.

CONCLUSION

Privacy-preserving data mining is an emerging field that studies the interactions between privacy and data mining following two main directions. The first direction seeks to evaluate the impact of the data mining process on privacy, and particularly the privacy of specific individuals. The second direction concerns how to accomplish the data mining process without disclosing unnecessary information regarding the data on which the mining process is performed. Several different algorithms have been developed in the last years that can be classified in three different broad families of privacy-preservation techniques. It is now time for the domain to reflect on what has been achieved, and more importantly on what should be achieved next. In particular, it seems important to find objective ways of comparing two different algorithms and the level of privacy they offer. This task is made harder by the complexity of defining the concept of privacy itself. Regarding this direction of research, a formal framework was proposed (Bertino, Fovino & Provenza, 2005) which allows to compare different privacy-preserving algorithms using criteria such as efficiency, accuracy,

scalability or level of privacy. Regarding the future trends, understanding how much the result of a data mining process reveals about the original dataset(s) and how to cope with malicious participants are two main challenges that lie ahead.

REFERENCES

Agrawal, D., & Aggarwal, C.C. (2001). On the design and quantification of privacy preserving data mining algorithms. In *Proceedings of the 20th ACM Symposium of Principles of Databases Systems*, 247–255.

Agrawal, R., & Srikant, R. (2000). Privacy-preserving data mining. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 439–450.

Agrawal, R., Srikant, R., & Dilys, T. (2005). Privacy preserving data OLAP. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 251–262.

Atallah, M. J., Bertino, E., Elmagarmid, A. K., Ibrahim, M., & Verykios, V. S. (1999). Disclosure limitations of sensitive rules. In *Proceedings of the IEEE Knowledge and Data Engineering Workshop*, 45–52.

Bertino, E., Fovino, I. N., & Provenza, L. P. (2005). A framework for evaluating privacy preserving data mining algorithms. *Data Mining and Knowledge Discovery*, 11(2), 121–154.

Chang, Y.-C., & Lu, C.-J. (2001). Oblivious polynomial evaluation and oblivious neural learning. In *Proceedings of Asiacrypt'01*, 369–384.

Chang, L., & Moskowitz, I .L. (2000). An integrated framework for database inference and privacy protection. In *Proceedings of Data and Applications Security*, 161–172.

Chaum, D., Crépeau, C., & Damgård, I. (1988). Multiparty unconditionally secure protocols. In *Proceedings of the 20th ACM Annual Symposium on the Theory of Computing*, 11–19.

Evfimievski, E., Gehrke, J. E., & Srikant, R. (2003). Limiting privacy breaches in privacy preserving data mining. In *Proceedings of the 22nd ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Databases Systems*, 211–222.

Feigenbaum, J., Ishai, Y., Malkin, T., Nissim, K., Strauss, M., & Wright, R. (2001). Secure multiparty computation of approximations. In *Proceedings of the 28th International Colloquium on Automata, Languages and Programming*, 927–938.

Fienberg, S. E., & McIntyre, J. (2004). Data swapping: variations on a theme by Dalenius and Reiss. In *Proceedings of Privacy in Statistical Databases*, 14–29.

Gambs, S., Kégl, B., & Aïmeur, E. (2007). Privacy-preserving boosting. *Data Mining and Knowledge Discovery*, 14(1), 131–170.

(An earlier version by Aïmeur, E., Brassard, G., Gambs, S., & Kégl, B. (2004) was published in *Proceedings of the International Workshop on Privacy and Security Issues in Data Mining* (in conjunction with PKDD'04), 51–69.)

Goldreich, O. (2004). *Foundations of Cryptography, Volume II: Basic Applications*. Cambridge University Press.

Kantarcioglŭ, M., & Vaidya, J. (2004). Privacy preserving naive bayes classifier for horizontally partitioned data. In *Proceedings of the Workshop on Privacy Preserving Data Mining* (held in association with The Third IEEE International Conference on Data Mining).

Kruger, L., Jha, S., & McDaniel, P. (2005). Privacy preserving clustering. In *Proceedings of the 10th European Symposium on Research in Computer Security*, 397–417.

Ishai, Y., Malkin, T., Strauss, M.J., & Wright, R. (2007). Private Multiparty Sampling and

Approximation of Vector Combinations. In *Proceedings of the 34th International Colloquium on Automata, Languages and Programming*, 243–254.

Laur, S., Lipmaa, H., & Mielikäinen, T. (2006). Cryptographically private support vector machines. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 618–624.

Lindell, Y., & Pinkas, B. (2000). Privacy preserving data mining. In Proceedings of Crypto'2000, 36–54.

(Extended version available in (2002) *Journal of Cryptology*, 15, 177–206.)

Meyerson, A., & Williams, R. (2004). On the complexity of optimal *k*-anonymity. In *Proceedings of the 23rd ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Databases Systems*, 223–228.

Sweeney, L. (2002). Achieving *k*-anonymity privacy protection using generalization and suppression. *International Journal on Uncertainty, Fuzziness, and Knowledge-based Systems*, 10(5), 571–588.

Vaidya, J., Clifton, C.W., & Zhu, Y.M. (2006). *Privacy Preserving Data Mining*. Advances in Information Security, Springer Verlag.

Verykios, V. S., Bertino, E., Fovino, I. N., Provenza, L. P., Saygin, Y., & Theodoridis, Y. (2004). State-of-the-art in privacy preserving data mining. *SIGMOD Record*, 3(1), 50–57.

Výborný, O. (2006). Privacy preserving data mining, state-of-the-art. *Technical Report from the Faculty of Informatics, Masaryk University Brno*.

KEY TERMS

ID3 (Iterative Dichotomiser 3): Learning algorithm for the induction of decision tree (Quinlan, 1986) that uses an entropy-based metric to drive the construction of the tree.

***k*-Anonymization:** Process of building a *k*-anonymized dataset in which each individual record is indistinguishable from at least *k*-1 other records within the dataset.

Conditional Entropy: Notion of information theory which quantifies the remaining entropy of a random variable Y given that the value of a second random variable X is known. In the context of privacy-preserving data mining, it can be used to evaluate how much information is contained in a randomized dataset about the original data.

Expectation-Maximization (EM): Family of algorithms used in statistics for finding maximum likelihood estimates of parameters in probabilistic models, where the model depends on unobserved latent variables.

Privacy Breach: Regarding the particular attribute of a particular record, a privacy breach can occur if the release of the randomized dataset causes a change of confidence regarding the value of this attribute which is above the *privacy breach level*.

Sanitization: Process of removing sensitive information from a dataset so that it satisfies some privacy constraints before it can be release publicly.

Secure Multiparty Computation: Area of cryptography interested in the realization of distributed tasks in a secure manner. The security of protocols can either be computational (for instance based on some cryptographic assumptions such as the difficulty of factorization) or unconditional (in the sense of information theory).

Semi-Trusted Party: Third party that is assumed to follow the instructions that are given to him but can at the same time "spy" on all the information that he sees during the execution of the protocol. A virtual trusted third party can be emulated by combining a semi-trusted third party together with cryptographic tools.

Data Mining and the Text Categorization Framework

Paola Cerchiello
University of Pavia, Italy

INTRODUCTION

The aim of this contribution is to show one of the most important application of text mining. According to a wide part of the literature regarding the aforementioned field, great relevance is given to the classification task (Drucker et al., 1999, Nigam et al., 2000).

The application contexts are several and multitask, from text filtering (Belkin & Croft, 1992) to word sense disambiguation (Gale et al., 1993) and author identification (Elliot and Valenza, 1991), trough anti spam and recently also anti terrorism. As a consequence in the last decade the scientific community that is working on this task, has profuse a big effort in order to solve the different problems in the more efficient way.

The pioneering studies on text categorization (TC, a.k.a. topic spotting) date back to 1961 (Maron) and are deeply rooted in the Information Retrieval context, so declaring the engineering origin of the field under discussion. Text categorization task can be briefly defined as the problem of assigning every single textual document into the relative class or category on the basis of the content and employing a classifier properly trained.

In the following parts of this contribution we will formalize the classification problem detailing the main issues related.

BACKGROUND

In a formal way, text categorization is the task of assigning a Boolean value to:

$$(d_j, c_i) \in D \times C$$

where D is a domain of documents and $C = [c_1, \ldots\ldots, c_{|C|}]$ is a set of predefined categories. A value of T assigned to (d_j, c_i) indicates a decision to file d_j (also called *positive example*) under c_i , and a value of F assigned to (d_j, c_i) indicates a decision not to file d_j (also called *negative example*) under c_i. In other words a function, able to approximate the relation between the set D of documents and the set C of categories, is needed by means of a classifier able to learn the main characteristic of each categories on the basis of the information contained in the documents.

We emphasize that the *c* categories are simply symbolic labels (authors' name, spam or not spam, different topics) and often no further information is available.

Regarding the relation between documents and categories two different formulations should be mentioned (Sebastiani, 2003):

- Given a document $d_j \in D$, all the categories under which it could be filed, are checked, (*document-pivoted categorization*-DPC);
- Given a category $c_i \in C$, all the documents that could be filed under it, are checked (*category-pivoted categorization*-CPC).

Actually the above distinction is more practical than theoretical, in any case it can be useful whenever C and D are not fully available from the beginning.

Now we want to refer two main approaches employed in the construction of a text classifier.

The first one is the *Knowledge Engineering*, born in '80, dealing with the manual construction of several decision rules, of type *if (some characteristic) then (category i-th)*. As the reader can simply infer this approach was too expensive either from a human or a time point of view. In fact those rules needed, not only to be thought about by an human expert of the relative topic, but also to be updated every time some constitutive elements change.

The ideal field for TC was found in *Machine learning* community that introduced the idea of a supervised classifier able to learn the key elements of each category on the basis of some preclassified documents. We underline that according to this formulation, the

wish is not simply constructing a classifier, but mainly creating a builder of classifier automatizing the whole process.

Another important aspect of every application of text mining, not only text categorization, is the transformation of a textual document in a analyzable database. The idea is to represent a document d_j with a vector of weights $d_j = (w_{1j},....,w_{|T|j})$ according to which T is the dictionary, that is the set of terms present at least once in at least *k* documents, and *w* measures the importance of the term. With the concept of term the field literature is usual to refer to a single word from the moment that this kind of representation produces the best performance (Dumais et al. 1998).

On the other side w_i is commonly identified with the count of the i-th word in the document d_j, or with some variation of it like *tf-idf* (Salton & Buckley, 1988).

The data cleaning step is a key element of a text mining activity in order to avoid the presence of highly language dependant words. That latter objective is obtained by means of several activities such as conversion to lower case, managing of acronym, removing of white space special characters, punctuation, stop words, handling of synonymous and also word stemming.

A specific word frequency gives us information about the importance of it in the document analyzed, but it does not mean that the number they appear with, is proportional to the importance in the description of the above document. What is typically more important, it is the proportion between the frequencies of words to each other in the text.

Even after all the activities of data cleaning presented, the number of relevant words

is still very high (in the order of 10^3), thereby features selection methods must be applied (Forman, 2003).

MAIN THRUST

Once defined the preprocessing aspects of a text categorization analysis, it is necessary to focus on one or more classifiers that will assign every single document to the specific category according to the general rule specified above:

- The first and oldest algorithm employed in the field under analysis is the Rocchio method (Joachims, 1997) belonging to the linear classifier class and based on the idea of a category profile (document prototype). A classifier built with such methodology rewards the proximity of a document to the centroid of the training positive examples and its distance from the centroid of the negative examples. This kind of approach is quite simple and efficient however, because of the introduction of the linear division of the documents space, as all the linear classifiers do, it shows a low effectiveness giving rise to the classification of many documents under the wrong category.
- Another interesting approach is represented by the memory based reasoning methods (Masand et al., 1992) that, rather than constructing and explicit representation of the category (like Rocchio method does), rely on the categories assigned to training documents that result more similar to the test documents. Unlike the linear classifiers, this approach doesn't linearly divide the space and therefore we are not expected to have the same problems seen before. However the weak point of such formulation resides in the huge time of classification due to the fact that it doesn't exist a real phase of training and the whole computational part is left to classification time.
- Among the class of non numeric (or symbolic) classifier, we mention decision trees (Mitchell, 1996) which internal nodes represent terms, branches are tests on weights for each words and finally leaves are the categories. That kind of formulation classifies a document d_j by means of a recursive control of weights from the main root, trough internal nodes, to final leaves. A possible training method for a decision tree consists in a 'divide and conquer' strategy according to which first, all the training examples should present the same label, and if it is not so, the selection of a term t_k is needed on the basis of which the document set is divided. Documents presenting the same value for t_k are selected and each class is put in a separated sub-tree. That process is repeated recursively on each sub-tree until every single leaf contains documents belonging to the same category. The key phase of this formulation relies on the choice of the term based on indexes like information gain or entropy.
- According to another formulation, the probabilistic one, the main objective is to fit $P(c_i/d_j)$ that

represents the probability for each document to belong to category c_i. This value is calculated on the basis of the Bayes theorem:

$$P(c_i \mid d_j) = \frac{P(c_i)P(d_j \mid c_i)}{P(d_j)}$$

where $P(d_j)$ represents the probability for a document, randomly chose, to present the vector d_j as its representation, and $P(c_i)$ to belong to the category c_i.

One of the most famous formulation of a probabilistic classifier is constituted by the Naïve Bayes one (Kim et al., 2000) that uses the independence assumption regarding the terms present in the vector d_j. As the reader can simply infer that assumption is quite debatable but surely comfortable and useful in calculations and it constitutes the main reason why the classifier is called Naïve:

- Great relevance is given also to the support vector machines (SVMs) model, introduced by Joachims in 1998 in the TC field and which origin is typically not statistic but relies on machine learning area. They were born as linear classifier and represent a set of related supervised learning methods used both for classification and regression scope. When used for classification, the SVM algorithm (Diederich et al., 2003) creates a hyperplane that separates the data into two (or more) classes with the maximum margin. In the simplest binary case, given training examples labelled either c_i or $\neg c_i$, a maximum-margin hyperplane is identified which splits the c_i from the $\neg c_i$ training examples, such that the distance between the hyperplane and the closest examples (the margin) is maximized. We underline the best hyperplane is located by a small set of training example, called support vectors.

The undisputed advantages of the aforementioned classifier are:

1. the capacity of manage dataset with huge number of variables (as typically is the case of TC) without requiring activity of variable selection.
2. the robustness to problems of overfitting.

The more important disadvantage of the technique, especially at the beginning of its application, was the big computational cost that anyway has been overcome with the development of more efficient algorithms.

We add that, because of the inadequacy of the linear approach, as already said before, the non linear version of SVMs is more appropriate in high-dimensional TC problems:

- We also report some notes about ensembles classifiers (Larkey & Croft, 1996), representing one of the more recent development in the research of efficient methods for TC.

The basic idea of that approach is simple: if the best performance of each classifier are put together, the final results should be considerably better. In order to get that objective, those classifiers must be combined properly, that is in the right order and applying the correct combination function.

From an operational point of view each classifier is built by the learner given different parameters or data and is run on the test instances getting finally a ``vote". The votes are then collected and the category with the greatest number of votes becomes the final classification.

We refer the main approaches:

1. Bagging (Breiman, 1996): The training set is randomly sampled with replacement so that each learner gets a different input. The results are then voted.
2. Boosting (Schapire et al., 1999): The learner is applied to the training set as usual, except that each instance has an associated weight and all instances start with an equal weight. Every time a training example is misclassified, a greater weight is assigned by the algorithm. The process is recursive and it is repeated until there are no incorrectly classified training instances or, alternatively, a threshold defining the maximum number of misclassified examples is established.

Research papers (Bauer & Kohavi, 1999) indicate that bagging gives a reliable (i.e. it can be applied in many domains) but modest improvement in accuracy; whereas boosting produces a less reliable but greater improvement in accuracy (when you can use it):

- We finally offer some notes about another interesting and recent approach, rooted in the fuzzy logic context. In order to classify a text is, of course, necessary to determine the main faced topic. However the complete and exhaustive definition of a topic, or in general of a concept, is not an easy task. In this framework, the fuzzy approach can be usefully employed, by means of fuzzy sets. They are sets containing measure of the degree of membership for each contained element. The values are contained in the range [0;1] and the larger the value, the more the element is supposed to be a member of the set. The principal advantage of this formulation lays in the possible specification of topics in terms of hierarchy. The main disadvantage is the inevitable degree of subjectivity in the values assignation.

FUTURE TRENDS

In classical contest the aforementioned classifiers shows performance related to the specific advantages and disadvantages of the methods. However, when the context of analysis changes and the number of categories is not fixed but it can vary randomly without knowing the exact proportion of the increase, those methods fulfil.

A typical example can be represented by the situation in which a new document, belonging to an unknown class, has to be classified and the classical classifiers are not able to detect the text as an example of a new category.

In order to overcome this problem, a recently proposed approach, employed in the author identification context (Cerchiello, 2006), is based on the combined employment of feature selection, by means of decision trees and Kruskal-Wallis test.

That methodology selects terms located both by the decision tree and the non parametric Kruskal-Wallis test (Conover, 1971) and characterized by distribution profiles more heterogeneous and different from each other so that they can reveal a typical and personal style of writing for every single author. This part of the analysis is useful to eliminate words used in a constant or semi-constant way in the different documents composed by every single author.

What proposed above can be evaluated in terms of correspondence analysis, a typical descriptive method able to plot in a bidimensional space rows and columns profiles, underlying the below relationship between observations and variables (in our case documents and words).

The resulting plot is useful to locate documents that can possibly belong to a new category not previously considered.

CONCLUSION

To conclude this contribution we summarize the key elements of the aforementioned research field. As we have already said among the several application of text mining, great relevance is given to text classification task The area literature has proposed in the last decade the employment of the most important classifiers, some rooted in purely statistical field, some other instead in the machine learning one. Their performance depends mainly on the constitutive elements of the learner and among them SVMs and ensembles classifiers deserve more attention.

The formulation of the problem changes when the number of categories can vary, theoretically, in random way: that is documents not always belong to a preclassified class.

In order to achieve the solution of the problem a feature selection activity is needed enriched by a visual representation of documents and words, obtained in terms of correspondence analysis.

REFERENCES

Bauer, E., & Kohavi, R. (1999). An empirical comparison of voting classification algorithms: Bagging, boosting and variants, *Machine Learning*, 36, 105-142.

Belkin N. J., & Croft W. B. (1992). Information filtering and information retrieval: two sides of the same coin?, *Communication ACM 35 International Conference on Research and Development in Information Retrieval*, 12, 29–38.

Breiman, L. (1996). Bagging predictors, *Machine Learning*, 24, 123-140.

Cerchiello, P.(2006). Statistical methods for classi-

fication of unknown authors. Unpublished doctoral dissertation, University of Milan, Italy.

Conover, W. J. (1971). *Practical nonparametric statistics*. Wiley, New York.

Diederich J.,Kindermann J., Leopold E. & Paass G. (2003). Authorship Attribution with Support Vector Machines, *Applied Intelligence*, *19*(1), 109-123.

Drucker, H., Vapnik V., & Wu D. (1999). Automatic text categorization and its applications to text retrieval, *IEEE Transactions on Knowledge and Data Engineering*, *10*(5), 1048 -1054.

Dumais, S. T., Platt, J., Heckerman, D., & Sahami, M. (1998). Inductive learning algorithms and representations for text categorization. In *Proceedings of CIKM-98, 7th ACM International Conference on Information and Knowledge Management* (Bethesda, MD, 1998), 148–155.

Elliot W., & Valenza R. (1991). Was the Earl of Oxford the true Shakespeare?. Notes and Queries, 38:501-506,.

Forman G. (2003). An Extenisve empirical study of feature selection metrics for text classification, *Journal of Machine Learning Research*, 3, 1289-1306.

Gale W. A., Church K. W. & Yarowsky D., (1993). A method for disambiguating word senses in a large corpus. *Comput. Human. 26*, 5, 415–439.

Gray A., Sallis P., & MacDonell S. (1997). Software Forensics: Extending Authorship Analysis Techniques to Computer Programs". In Proc. 3rd Biannual Conf. Int. Assoc. of Forensic Linguists (IAFL'97), 1-8.

Joachims T. (1998). Text categorization with support vector machines: learning with many relevant features, In: *Proceedings of 10th European Conference on Machine Learning* (Chemnitz, Germany, 1998), 137 142

Joachims T. (1997). A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization, In: *Proceedings of 14th International Conference on Machine Learning* (Nashville, TN, 1997), 143 -151.

Kim Y. H.,Hahn S. Y., & Zhang B. T. (2000). Text filtering by boosting naive Bayes classifiers, In *Proceedings of 23rd ACM International Conference on Research and Development in Information Retrieval* (Athens, Greece, 2000), 168 -175.

Larkey L. S., & Croft W. B. (1996). Combining classifiers in text Categorization. In *Proceedings of 19th ACM International Conference on Research and Development in Information Retrieval* (Zurich, Switzerland) 289-297.

Maron M., (1961). Automatic Indexing: an experimental inquiry, *Journal of the Association for Computing Machinery*, 8, 404-417.

Masand B., Linoff G., & Waltz D., (1992). Classifying news stories using memory-based reasoning. In *Proceedings of 15th ACM International Conference on Research and Development in Information Retrieval* (Copenhagen, Denmark, 1992), 59–65.

Mitchell T.M., (1996). *Machine Learning.* McGraw Hill, New York, NY.

Nigam K., Mccallum A. K., Thrun S., & Mitchell T. M., (2000). Text classification from labeled and unlabeled documents using EM. *Journal of Machine Learning Research, 39*(2/3), 103-134.

Salton, G., & Buckley C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing Managing, 24*(5), 513–523.

Schapire, R., (1999). A brief introduction to boosting, In: *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence,* 1401-1405.

Sebastiani F., (2003). Machine learning in automated text categorization. *ACM Computing Surveys 34*(1), 1-47.

KEYTERMS

Correspondence Analysis: It is a method of factoring categorical variables and displaying them in a property space which maps their association in two or more dimensions. It is often used where a tabular approach is less effective due to large tables with many rows and/or columns. It uses a definition of chi-square distance rather than Euclidean distance between points, also known as "profile points". A point is one of the values (categories) of one of the discrete variables in the analysis.

Decision Trees: They are also known as classification tree and constitute a good choice when the task of

the analysis is classification or prediction of outcomes and the goal is to generate rules that can be easily understood and explained. They belong to the family of non parametric models and in text mining field they represent trees in which, internal nodes are labelled by terms, branches departing from them are labelled by tests on the weight that the term has in the test document, and finally leafs represent category.

Information Retrieval: The term "information retrieval" was coined by Calvin Mooers in 1948-50 and it represents the science of searching for information in documents, searching for documents themselves or searching for metadata which describe documents. IR derives from the fusion of interdisciplinary fields such as information architecture, human information behavior, linguistics, semiotics, information science and computer science.

Knowledge Engineering: It is the process of building intelligent Knowledge-based systems and is deeply rooted in other areas such as databases, datamining, expert systems and decision support systems.

Kruskal-Wallis Test: It is the non parametric version of ANOVA analysis and it represents a simple generalization of the Wilcoxon test for two independent sample. On the basis of K independent samples n_1, …, n_k , a unique big sample is created by means of fusion of the originals k samples. The above result is ordered from the smaller sample to the bigger one, and the rank is assigned to each one. Finally R_i is calculated, that is the mean of the ranks of the observations in the ì-th sample. The statistic is:

$$KW=\left[\frac{12}{N(N+1)}\sum_{j=1}^{k}\frac{R_j^2}{n_j}\right]-3(N+1)$$

and the null hypothesis, that all the k distributions are the same, is rejected if:

$KW > X^2_{ü(k-1)}$.

Machine Learning: It is an area of artificial intelligence concerned with the development of techniques which allow computers to "learn". In particular it is a method for creating computer programs by the analysis of data sets.

Text Mining: It is a process of exploratory data analysis applied to data in textual format that leads to the discovery of unknown information, or to answers to questions for which the answer is not currently known.

Data Mining Applications in Steel Industry

Joaquín Ordieres-Meré
University of La Rioja, Spain

Manuel Castejón-Limas
University of León, Spain

Ana González-Marcos
University of León, Spain

INTRODUCTION

The industrial plants, beyond subsisting, pursue to be leaders in increasingly competitive and dynamic markets. In this environment, quality management and technological innovation is less a choice than a must. Quality principles, such as those comprised in ISO 9000 standards, recommend companies to make their decisions with a based on facts approach; a policy much easily followed thanks to the all-pervasive introduction of computers and databases.

With a view to improving the quality of their products, factory owners are becoming more and more interested in exploiting the benefits gained from better understanding their productive processes. Modern industries routinely measure the key variables that describe their productive processes while these are in operation, storing this raw information in databases for later analysis. Unfortunately, the distillation of useful information might prove problematic as the amount of stored data increases. Eventually, the use of specific tools capable of handling massive data sets becomes mandatory.

These tools come from what it is known as 'data mining', a discipline that plays a remarkable role at processing and analyzing massive databases such as those found in the industry. One of the most interesting applications of data mining in the industrial field is system modeling. The fact that most frequently the relationships amongst process variables are nonlinear and the consequent difficulty to obtain explicit models to describe their behavior leads to data-based modeling as an improvement over simplified linear models.

BACKGROUND

The iron and steel making sector, in spite of being a traditional and mature activity, strives to approach new manufacturing technologies and to improve the quality of its products. The analysis of process records, by means of efficient tools and methods, provides deeper knowledge about the manufacturing processes, therefore allowing the development of strategies to cut costs down, to improve the quality of the product, and to increase the production capability.

On account of their anticorrosive properties, the galvanized steel is a product experiencing an increasing demand in multiple sectors, ranging from the domestic appliances manufacturing to the construction or automotive industry. Steel making companies have established a continuous improvement strategy at each of the stages of the galvanizing process in order to lead the market as well as to satisfy the, every time greater, customer requirements (Kim, Cheol-Moon, Sam-Kang, Han, C. & Soo-Chang, 1998; Tian, Hou & Gao, 2000; Ordieres-Meré, González-Marcos, González & Lobato-Rubio, 2004; Martínez-de-Pisón, Alba, Castejón & González, 2006; Pernía-Espinoza, Castejón-Limas, González-Marcos & Lobato-Rubio, 2005).

The quality of the galvanized product can be mainly related to two fundamental aspects (Lu & Markward, 1997; Schiefer, Jörgl, Rubenzucker & Aberl, 1999; Tenner, Linkens, Morris & Bailey, 2001):

- As to the anticorrosive characteristics, the quality is determined by the thickness and uniformity of the zinc coat. These factors basically depend on

the base metal surface treatment, the temperature of its coating and homogenization, the bath composition, the air blades control and the speed of the band.

- As to the steel properties, they mainly depend on the steel composition and on the melting, rolling and heat treatment processes prior to the immersion of the band into the liquid zinc bath.

MAIN FOCUS

Data mining is, essentially, a process lead by a problem: the answer to a question or the solution to a problem is found through the analysis of the available records. In order to facilitate the practice of data mining, several standardization methods have been proposed. These divide the data mining activities in a certain number of sequential phases (Fayyad, Piatetsky-Shapiro, & Smyth, 1996; Pyle, 1999; Chapman, Clinton, Kerber, Khabaza, Reinartz, Shearer, & Wirth, 2000). Nevertheless, the names and contents of these phases slightly differ. The same general concept is present in every proposal: first the practitioner becomes familiar with the problem and the data set; later on, the data is adapted to suit the needs of the algorithms; then, models are built and tested; finally, newly acquired knowledge provides support for an enhanced decision making process.

Nowadays, nonlinear modeling comprises important techniques that have reached broad applicability thanks to the increasing speed and computing capability of computers. Genetic algorithms (Goldberg & Sastry, 2007) , fuzzy logic (Wang, Ruan & Kerre, 2007) and cluster analysis (Abonyi & Feil, 2007), to name a few, could be mentioned amongst these techniques. Nevertheless, if there is a simple and efficient technique that has made inroads into the industry, that definitely is the artificial neural network.

Amongst the currently existing neural networks, one of the most adequate for defining systems and processes by means of their inputs and outputs is the multilayer perceptron (MLP), which can be considered a universal function approximator (Funahashi, 1989; Hornik, Stinchcombe. & White, 1989); a MLP network with one hidden layer only needs a large enough number of nonlinear units in its hidden layer to mimic any type of function or continuous relationship amongst a group of input and output variables.

The following lines show the successful application of the MLP network, along with other data mining tools, in the modeling of a hot dip galvanizing line whose existing control systems were improved.

Estimation of The Mechanical Properties of Galvanized Steel Coils

The core of the control engineering strategies relies on the capability of sensing the process variables at a rate high enough so as to be able to act in response to the observed behavior. 'Closed loop' control strategies, as they are commonly referred to, are based on acting over the commanding actions to correct the differences amongst the observed and expected behavior. In that scenario, control engineers can choose amongst many effective and well known approaches.

Unfortunately, nowadays the mechanical properties of the galvanized steel coils (yield strength, tensile strength and elongation) cannot be directly measured. Instead, they have to be obtained through slow laboratory tests, using destructive methods after the galvanizing process. The lack of online (fast enough) measures compels the control engineers to adopt an 'open loop' control strategy: the estimation of the best commanding actions in order to provide the desired outputs.

As it has already been mentioned, the mechanical properties of the galvanized steel coils change during the manufacturing processes: from the steel production processes that determine their chemical composition, to the very last moment in which it becomes a finished product, either in the shape of galvanized coils or flat plates.

Ordieres-Meré, González-Marcos, González & Lobato-Rubio (2004) proposed a neural network model based on the records of the factory processes in operation. The model was able to predict the mechanical properties of the galvanized steel not only with great accuracy, which is mandatory to improve the quality of the products, but also fast enough, what is critical to close the loop and further improve the quality thanks to a wider range of more complex strategies.

This particular data-based approach to system modeling was developed to satisfy the needs of ARCELOR SPAIN. Following data mining practice standards, the 1731 samples of the data set were analyzed and prepared by means of visualization techniques (histograms, scatter plots, etc.) and projection techniques (Sammon projection (Sammon, 1969) and principal components

Table 1. Test samples mean relative error for each mechanical property and cluster analysed.

Mechanical property	Cluster1	Cluster2
Yield strength (MPa)	3.19 %	2.60 %
Tensile strength (MPa)	0.88 %	1.82 %
Elongation (%)	3.07 %	4.30 %

analysis, PCA (Dunteman, 1989)); later on, the outliers were identified and the data set was split according to the cluster analysis results; then, a MLP network was selected amongst a set of different candidates.

In this case, only two out of the three detected clusters contained a large enough number of samples to make posible a later analysis. One of the clusters, 'cluster 1', comprised 1142 samples, while the other, 'cluster 2', contained 407 samples. For these two clusters, a set of MLP networks was trained. These networks featured an input layer with 17 inputs, a hidden layer with a variable number of neurons ranging from 3 to 20, and an output layer with a single output neuron. The selected learning algorithm was backpropagation with weight decay for preventing the network from overfitting.

The obtained results were pretty good and the test pattern mean relative error (see Table I) was within tolerance, showing the feasibility of online estimations. The advantages of these models in terms of economic savings, derived from the online forecast of these mechanical characteristics, are admirable.

In the past, the mechanical characteristics were only measured at discrete points. Currently, this neural model provides a continuous map of the estimates of the mechanical characteristics of the coils, thus identifying which portions comply with customer requirements. Instead of reprocessing the whole coil, the rejected portion can be removed from the coil with the consequent economic saving.

Modeling the Speed of the Steel Strip in the Annealing Furnace

Heat treatment is a key process to obtain the desired properties of the band and one that greatly determines the good adhesion of the zinc coating. Before a neural model was built, the temperature of the band during the heat treatment was controlled only by acting over the furnace temperature. Martínez-de-Pisón (2003) proposed a more effective temperature control by means of changing the speed of the band as well. Following this approach, Pernía-Espinoza, Castejón-Limas, González-Marcos & Lobato-Rubio (2005) made a step forward improving ARCELOR SPAIN annealing process quality by means of a robust model of the strip speed with which more adequate temperatures were obtained.

The data set supporting the model contained a small fraction of samples, (1040 out of 30920 samples) with values of speed outside the normal operation range. They were due to transient regimes, such as those caused by the welding of a coil or the addition of an atypical coil (with unusual dimensions) to the line. In this situation, an abrupt reduction of the speed stands to reason, making advisable the use of a neural model to express the relationships amongst the temperature and the speed.

In order to diminish the influence of the outliers, a robust approach was considered. The robust neural network proposed by Liano (1996) was selected because of its high breakdown point along with its high efficiency against normally distributed errors. The MLP networks training was perform by the conjugated gradient Fletcher-Reeves method.

The results on the test pattern (20% of the samples: 6184 patterns) showed a 4.43% relative mean error, against a 5.66% obtained with a non-robust neural network. That reflected the good behavior of the robust model and proved right the feasibility of a much more effective control of the coils temperature. Moreover, the robust model might be useful while planning the operation of the line, sorting the queue of coils to be processed, or forecasting the speed conditions during transients.

Skin-Pass Artificial Lock

The development of an artificial lock in the skin-pass (González-Marcos, Ordieres-Meré, Pernía-Espinoza & Torre-Suárez, to appear), a phase of the galvanizing process that, inter alia, endows the desired roughness to the surface of the band, illustrates the spawn of new ideas and perceptions that the data mining process bears.

The idea of creating this lock arouse as a consequence of the good results obtained in the prediction of the mechanical properties of the galvanized steel coils. The faulty labeling of the coils is a problem that, although uncommon, used to have severe consequences. Under

these circumstances, a coil with a wrong label used to be treated according to false parameters, thus ending up with hidden and unexpected mechanical properties. Failing to comply with the customer requirements can cause important damages to the customers' machinery, such as fractures while handling a material that is way stronger that expected. In order to avoid this kind of mistakes, an artificial lock was developed in the skin-pass. The lock is a model able to provide online forecast of the elongation of the coils in the skin-pass using the manufacturing and chemical composition data. Thus, a significant difference amongst the expected and observed elongation would warn about a probable mistake during the labeling. Such warning causes the coil to be removed and further analyzed in order to identify the cause of such difference.

The results obtained (Table II) show the goodness of the developed models. Besides, Their utility to solve the problem that originated the study was proved with coils whose incorrect chemical composition was previously known. In these cases, the observed and estimated elongation were significantly different, with errors ranging from -0.334 to -1.224, what would identify, as desired, that the coils under process were wrongly labeled.

The training of the MLP network was performed in a computer with two Xeon 64 processors running at 2.4 GHz, with 8 Gb of RAM memory and Linux operative system. As an example, the training of a network with 26 input neurons, 35 hidden neurons and a single output neuron took 438 minutes, using 3580 training patterns and 1760 validation patterns and running 50000 cycles.

FUTURE TRENDS

Data mining opens a world of new opportunities for many industries. The iron and steel making sector has already taken advantage of a few of the tools the data mining has to offer, successfully applying them in an effective manner at otherwise very hard problems. But the box has just been opened, and many more advanced tools and techniques are on their way to come. Fortunately, the industry has already recognized the benefits of data mining and is eager to exploit its advantages.

CONCLUSION

Data mining is useful wherever data can be collected. Of course, in some instances, cost/benefit calculations might show that the time and effort of the analysis is not worth the likely return. Obviously, the idea of 'Data Mining wherever and whenever' is not advisable. If an application requires a model that can be provided by classical tools, then that is preferable insofar as this procedure is "less energy consuming" than those linked to DM methodologies.

We have presented various types of problems where data management can yield process improvements and where usually they are not so frequent due to the non-direct way of implementation.

In spite of these difficulties, these tools can provide excellent results based on a strict methodology like CRISP-DM (Chapman, Clinton, Kerber, Khabaza, Reinartz, Shearer, & Wirth, 2000) in combination with an open mind.

Data mining can yield significant, positive changes in several ways. First, it may give the talented manager a small advantage each year, on each project, with each customer/facility. Compounded over a period of time, these small advantages turn into a large competitive edge.

Table 2. Mean absolute error for each steel class analysed.

Steel class	Total samples	Test samples	Test mean absolute error
0	1022	153	0.0874
1	6282	942	0.0712
2	2003	300	0.0652
4	1484	222	0.1285

REFERENCES

Abonyi, J & Feil, B. (2007) *Cluster analysis for data mining and system identification.* Birkhäuser Basel.

Bellman, R. (1961). *Adaptive control processes: A guided tour*. New Jersey, USA: Princeton University Press.

Bloch, G.; Sirou, F.; Eustache, V. & Fatrez, P. (1997). Neural intelligent control for a steel plant. *IEEE Transactions on Neural Networks*, *8*(4), 910-918.

Chapman, P.; Clinton, J.; Kerber, R.; Khabaza, T.; Reinartz, T.; Shearer, C. & Wirth, R. (2000). CRISP-DM 1.0: Step-by-step data mining guide. *CRISP-DM consortium / SPSS Inc*.,. Retrieved from http://www.crisp-dm.org.

Dunteman, G.H. (1989). *Principal components analysis*. Newbury Park, CA: Sage Publications.

Fayyad, U.; Piatetsky-Shapiro, G. & Smyth, P. (1996). Knowledge discovery and data mining: Towards a unifying framework. In E. Simoudis, J. Han & U. Fayyad (Eds.), *Proceeding of The Second International Conference on Knowledge Discovery and Data Mining (KDD'96)* (pp. 82-88). Menlo Park: AAAI Press.

Funahashi, K. (1989). On the approximate realization of continuous mapping by neural networks. *Neural Networks*, *2*, 183-192.

Goldberg, D. & Sastry, K. (2007) *Genetic Algorithms: The Design of Innovation.* Springer

González-Marcos, A.; Ordieres-Meré, J.B.; Pernía-Espinoza, A.V & Torre-Suárez, V. (in print). Desarrollo de un cerrojo artificial para el skin-pass en una línea de acero galvanizado por inmersión en caliente. *Revista de Metalurgia*.

Hornik, K.; Stinchcombe, M. & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*(5), 359-366.

Kim, Y. I.; Cheol-Moon, K.; Sam-Kang, B.; Han, C. & Soo-Chang, K. (1998). Application of neural network to the supervisory control of a reheating furnace in the steel industry. *Control Engineering Practice*, *6*, 1009-1014.

Liano K. (1996). Robust Error Measure for Supervised Neural Network Learning with Outliers. *IEEE Transactions on Neural Networks*, 7(1), 246-250.

Lu, Y.Z. & Markward, S.W. (1997). Development and application of an integrated neural system for an HDCL. *IEEE Transactions on Neural Networks*, *8*(6), 1328-1337.

Ordieres-Meré, J.B.; González-Marcos, A.; González, J.A. & Lobato-Rubio, V. (2004). Estimation of mechanical properties of steel strips in hot dip galvanizing lines. *Ironmaking and Steelmaking*, *31*(1), 43-50.

Martínez-de-Pisón, F.J. (2003). *Optimización mediante técnicas de minería de datos del ciclo de recocido de una línea de galvanizado*. Doctoral dissertation, University of La Rioja, Spain.

Martínez-de-Pisón, F. J.; Alba, F.; Castejón, M. & González, J.A. (2006). Improvement and optimisation of a hot dip galvanizing line using neural networks and genetic algorithms. *Ironmaking and Steelmaking*, *33*(4), 344-352.

Pernía-Espinoza, A.V.; Castejón-Limas, M.; González-Marcos, A. & Lobato-Rubio, V. (2005). Steel annealing furnace robust neural network model. *Ironmaking and Steelmaking*, *32*(5), 418-426.

Pyle, D. (1999). *Data Preparation for Data Mining.* San Francisco, USA: Morgan Kaufmann Publishers.

Sammon, J.W. (1969). A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, *18*(5), 401-409.

Schiefer, C.; Jörgl, H. P.; Rubenzucker, F. X. & Aberl, H. (1999). Adaptive control of a galvannealing process with a radial basis function network'. *Proceedings of the 14th IFAC World Congress, Vol. 0* (pp. 61-66). Beijing, PR China.

Tenner, J.; Linkens, D. A.; Morris, P.F. & Bailey, T. J. (2001). Prediction of mechanical properties in steel heat treatment process using neural networks. *Ironmaking and Steelmaking*, *28*(1), 15-22.

Tian, Y.C.; Hou, C.H. & Gao, F. (2000). Mathematical modelling of a continuous galvanizing annealing furnace. *Developments in Chemical Engineering & Mineral Processing*, *8*(3/4), 359-374.

Wang, P.P, Ruan, D & Kerre, E.E. (2007) *Fuzzy logic: A spectrum of theoretical & practical issues (studies in fuzziness and soft computing*). Springer.

KEY TERMS

Artificial Neural Network: A set of connected artificial neurons.

Artificial Neuron: A mathematical model that applies a function to a linear combination of the input variables in order to provide an output value.

Closed Loop Control System: A system that utilizes feedback as a mean to act over the driving signal according to the differences found amongst the observed and expected behavior.

Multilayer Perceptron: A specific kind of artificial neural network whose neurons are organized in sequential layers and where the connections amongst neurons are established only amongst the neurons of two subsequent layers.

Learning Algorithm: In the artificial neural network area, the procedure for adjusting the network parameters in order to mimic the expected behavior.

Open loop Control System: A system that does not rely on feedback to establish the control strategies.

Outlier: An observation that does not follow the model or pattern of the majority of the data.

Overfitting: Fitting a model to best match the available data while loosing the capability of describing the general behavior.

Robust: Not affected by the presence of outliers.

Steady-state Regime: Status of a system that stabilized.

Transient Regime: Status of a system that is changing from one steady-state regime to another.

Data Mining Applications in the Hospitality Industry

Soo Kim
Montclair State University, USA

Li-Chun Lin
Montclair State University, USA

Yawei Wang
Montclair State University, USA

INTRODUCTION

Some people say that "success or failure often depends not only on how well you are able to collect data but also on how well you are able to convert them into knowledge that will help you better manage your business (Wilson, 2001, p. 26)." It is said the $391 billion restaurant industry generates a massive amount of data at each purchase (Wilson, 2001), and once collected, such collected data could be a gigantic tool for profits. In the hospitality industry, knowing your guests in terms of where they are from, how much they spend money, and when and what they spend it can help hospitality managers formulate marketing strategies, enhance guest experiences, increase retention and loyalty and ultimately, maximize profits. Data mining techniques are suitable for profiling hotel and restaurant customers due to their proven ability to create customer value (Magnini, Honeycutt, & Hodge, 2003; Min, Min & Emam, 2002). Furthermore, if the hospitality industry uses such data mining processes as collecting, storing, and processing data, the industry can get strategic competitive edge (Griffin, 1998). Unfortunately, however, the hospitality industry and managers are behind of using such data mining strategies, compared to the retail and grocery industries (Bogardus, 2001; Dev & Olsen, 2000). Therefore, there is a need for learning about such data mining systems for the hospitality industry. The purpose of this paper is to show the applications of data mining systems, to present some successes of the systems, and, in turn, to discuss some benefits from the systems in the hospitality industry.

BACKGROUND

Simply speaking, data mining is the use of the data from the warehouse to discover unpredictable patters, trends and threats through multidimensional analysis or on-line analytical processing, or OLAP (Peacock, 1998; Ross, 1997). The hospitality industry is known as a highly customer-centered business and accumulates large amounts of customer data from central reservation systems (CRS), property management system (PMS), point-of-sale (POS), and guest loyalty program databases. Therefore, data mining application can play a huge role in the hospitality industry (Monash, 2006). The volume of guest information collected via electronic transactions is greater than what humans can easily manage without the aid of technology (Magnini, Honeycutt, & Hodge, 2003). Data-warehousing and data-mining technologies can easily handle large and complex databases and assist hoteliers and restaurateurs in predicting future customers' behaviors, designing marketing campaigns, supporting market analysis, evaluating and refining loyalty programs, creating strategies, and conducting trends analysis (Buchthal, 2006; Singh & Kasavana, 2005; Magnini, Honeycutt, & Hodge, 2003; Min, Min & Emam, 2002; Rowe, 1999).

MAIN FOCUS

Success Story in the Restaurant Industry

As there has been a request for using the data information for hotels and restaurants to survive in a competitive world, Atlanta-based AFC Enterprises and Boston Market have achieved their goals through a data mining process. Using such data mining technique, they eventually try to cultivate more loyal customers, retain core customers, and maximize profits (Ross, 1997).

For instance, AFC first gathers customer-purchase data by item, day, and combination of them. Then, it stores the data in a data warehouse using AIX RISC 6000 hardware and a mix of software tools. While doing such process, the management clearly learns about what factors affect profits and promotions, and who loyal customers are. AFC also uses data mining techniques to manage cash management, inventory management, and customer relationship management (CRM). More successful exams are Red Robin International, a 135-unit casual-dining chain based in Englewood, Colorado and Pizzeria Uno. In additions, these two companies use data mining technique for changing menu or for giving instant and right information to the marketing department for promotions and for customer satisfaction.

Another case for successfully implementing data mining systems is a 13-unit Italian dinner-house, Louise's Trattoria. Luoise's Trattoria, which was bankruptcy in late 1997 and now successfully come back to the business, has been using analysis of credit-card transactions and customer-purchase data, part of data mining. President of this restaurant chain strongly believed that such credit-card transactions could give him more accurate information regarding his customers than traditional guest surveys did (Liddle, 2000; Magnini, Honeycutt, & Hodge, 2003).

Clever Ideas Inc. provides CLICK/Valued Member, the system that uses data collected from credit-card transactions and POS systems to better identify, track, reward, and communicate frequent diners. After having complete customers' information from credit-card transaction and POS systems, the company also sends direct-mail, a quarterly newsletter offering special events or menu changes, and a thank-you email to the frequent diners. Not only customers like such system, but employees like the system, because the employees can get tipped on the full amount (Waters, 2000).

Success Story in the Hotel Industry

Harrah's Entertainment (an official name of the company), a leading company in the hotel and gaming industry, has gained great success from a customer-service oriented strategy centered on data-mining techniques. In 1997, Harrah's hotels and casinos introduced a loyalty-card program "Total Rewards", which tracked customers' purchasing activities and provided incentives to visit Harrah's properties throughout the country (Loveman, 2003). Harrah's tracked customers' purchasing and gaming patterns and then provided its customers with most-effective incentives. Through data mining techniques, Harrah's developed quantitative models to predict lifetime value of its customer and used them to center marketing and service delivery programs in increasing customer loyalty (Bligh & Turk, 2004; Freedman, 2003).

Harrah's also discovered that 26 percent of its customers accounted for 82 percent of company revenue (Magnini, Honeycutt, & Hodge, 2003). Harrah's continues integrating data across properties, developing models, mining the data, and running marketing and service delivery programs and propels it to be the leading position within gaming industry (Bligh & Turk, 2004).

Web-Enabled Customer Relationship Management

Swift (2001, p. 12) defined customer relationship management (CRM) as "enterprise approach to understanding and influencing customer behavior through meaningful communications in order to improve customer acquisition, customer retention, customer loyalty, and customer profitability." CRM evolves from direct sales to mass marketing, target marketing and then to customer relationship marketing with an emphasis that marketing and CRM are inseparable (Ling & Yen, 2001). CRM is used to identify individual preferences, demographics, psychographic profiles and other measures to predict future behaviors and customize marketing efforts (Hendler & Hendler, 2004). In addition, CRM often identifies valued customers who repeatedly purchase a great deal of hotel and restaurant services (Min, Min & Emam, 2002).

It is clear that data mining plays a critical role in CRM systems and helps transform customer data into useful information and knowledge (Ngai, 2005). For

example, Townsend Hotel in Birmingham, MI and the Red Mountain Spa in St. George, UT have used CRM and e-marketing services, Digital Alchemy, to understand their frequent guests and to provide them with excellent guest services. If both hotels know guests' email address, for instance, they send a confirmation email and a follow-up email to the guests. They also use CRM as a marketing tool to track special guests who got married in the hotels to offer an anniversary stay (Ostrowski, 2005).

The information can be used to tailor marketing strategies more effectively to specific customers. Web-enabled CRM systems have gained popularity in the hospitality industry by cutting the cost, time, and energy normally associated with database marketing (Oliva, 2001). What makes Web-based CRM systems so ideal for such hospitality operations is the data stored in one common place – the Internet, instead of individual property management systems (PMSs). Most Web-based CRM systems work as a data warehouse, pulling data from central reservation systems (CRSs) and PMSs. In addition, Web-based CRM is so accessible that every hotelier and restaurateur only needs a password to log on the Internet and run reports according to behavior segmentation, customer profiles and geo-demographic statistics (Oliva, 2001). In the example of Townsend Hotel and the Red Mountain Spa, the hotels' property management system (PMS) and e-marketing service provider Digital Alchemy have a direct link so that CRM mines data from the PMS and sends reservation emails (Ostrowski, 2005).

Another example of the use of CRM in the hospitality industry is that Wyndham International, one of the five largest U. S. based hotel chains, has used its membership-based CRM initiative: Wyndham ByRequest, which includes the web-site, the preferences databases, and integrated operational systems (e.g., PMS). When a guest provides his/her information in ByRequest, it completes guest profile including guest's general and contact information, room preferences, credit-card and express check-in/check-out preferences, airline frequent-flyer preferences, personal interests, and beverage preference for a consistent level of personalized service (Piccoli, O'Connor, Capaccioli, & Alvarez, 2003).

FUTURE TRENDS

Although the hospitality industry plunged into the world of data mining later than the traditional retail industry, the hospitality industry's use of data mining will catch up the retail industry's soon (Ross, 1997). It is obvious that there have been such successful trends in the hotel and restaurant businesses and more successes can be found at Dunkin's Brands, Logans' Roadhouse, Buffalo Wild Wings, McAlister's Deli, Papa Murphy's Take 'n' Bake Pizza, IHOP (Prewitt, 2007). Based on the review from the previous sections regarding successful stories of using data mining, more hotels and restaurants should deploy data mining engines in their work sites. In addition, cooperation with food and beverage manufacturers, such as beer, wine and spirits, will be helpful for the industry, since such manufacturers also have been looking for reliable customer purchasing databases (Bogardus, 2001).

CONCLUSION

As can be presented in the previous sections, the ultimate goal of data mining is strong customer relationship in the hospitality industry. In addition, as defined, customer relationship management (CRM) stresses maximizing benefits for the business (Ling & Yen, 2001; Piccol et al., 2003). This technique can be especially valuable for the hospitality industry, given the fact that hospitality companies often miss customer demographics or psycho-graphics. Such data mining engines can be useful and helpful tools for any hotel and restaurant companies (Monash, 2006; Murphy, Hofacker, & Bennett, 2001). Although several successful examples have been mentioned in this paper, more companies should investigate benefits of data mining systems and received the benefits from the systems.

REFERENCES

Bligh, P. & Turk, D. (2004). Cashing In on customer loyalty. *Customer Relationship Management, 8*(6), 48-51.

Borgardus, C. (2001, October 22). One for all: Foodservice needs industry wide database for maximum revenue growth, *Nations' Restaurant News*, 20-22.

Buchthal, K. (2006, March 1). Return engagements, *Restaurants & Institutions*, 65-66.

Dev, C.S. & Olsen, M.D. (2000). Marketing challenges for the next decade, *Cornell Hotel and Restaurant Administration Quarterly, 41*(1), 41-47

Freedman, R. (2003). Helping clients value IT investments, *Consulting to Management, 14*(3), 33- 39.

Griffin, R. K. (1998). Data warehousing, *Cornell Hotel and Restaurant Administration Quarterly*, 39 (4), 35.

Hendler, R. & Hendler, F. (2004). Revenue management in fabulous Las Vegas: Combining customer relationship management and revenue management to maximize profitability, *Journal of Revenue and Pricing Management, 3*(1), 73-79.

Liddle, A. J. (2000, October, 30). Causal-dining chain's mining of customer data spurs change, management indicates, *Nation's Restaurant News*, p. 52.

Ling, R. & Yen, D.C. (2001). Customer relationship management: An analysis framework and implementation strategies, *Journal of Computer Information Systems, 41*(3), 82-97.

Loveman, G. (2003). Diamonds in the data mine, *Harvard Business Review, 81*(5), 109-113.

Magnini, V.P., Honeycutt, E.D. & Hodge, S.K. (2003). Data mining for hotel firms: Use and limitations, *Cornell Hotel and Restaurant Administration Quarterly, 44*(2), 94-105.

Min, H., Min, H. & Eman, A. (2002). A data mining approach to developing the profiles of hotel customers, *International Journal of Contemporary Hospitality Management, 14*(6), 274-285.

Monash, C.A. (2006, September 11). Data mining ready for a comeback, *Computerworld,* p. 45.

Murphy, J., Hofacker, C. F., & Bennett, M. (2001). Web-site-generated market-research data: Tracing the tracks left behind by visitors, *Cornell Hotel and Restaurant Administration Quarterly*, 41 (1), 82-91105.

Ngai, E.W.T. (2005). Customer relationship management research (1992-2002): An academic literature review and classification, *Marketing Intelligence & Planning, 23*(6), 582-605.

Oliva, R. (2001, March). Getting connected as database marketing becomes more complex, hotels turn to the Web to centralize and simplify the process, *Hotels*, 89-92.

Ostrowski, C. (2005, October). Hotels rely on CRM Technology to build relationships with repeat guests, *Hotel Business*, 12-13.

Peacock, P. R. (1998). Data mining in Marketing: Part I, *Marketing Management,* Winter, 9-18.

Piccoli, G, O'Connor, P, Capaccioli, C., & Alvarex, R. (2003). Customer relationship management – A driver for change in the structure of the U.S. lodging industry, *Cornell Hotel and Restaurant Administration Quarterly*, 44 (4), 61-73.

Prewitt, M. (2007, April 30). Customer data mining aids in quests for lucrative new sites, *Nation's Restaurant News*, p. 1.

Ross, J. R. (1997, July 14). Beyond the warehouse: Data mining sharpens competitive edge, *Nation's Restaurant News*, 73-76.

Rowe, M. (1999, July 1). Data mining: What data can do for you? *Lodging Hospitality, 55*(8), p. 40.

Singh, A.J. & Kasavana, M.L. (2005). The impact of information technology on future management of lodging operations: A Delphi study to predict key technological events in 2007 and 2027, *Tourism and Hospitality Research,* 6(1), 24-37.

Swift, .R.S. (2001). *Accelerating customer relationships using CRM and relationship technologies*, Prentice-Hall, PTR, Upper Saddle River, NJ.

Waters, C. D. (2000, September 4). Operators hope loyalty program blending high tech, direct mail CLICKs, *Nation's Restaurant News*, p. 20.

KEY TERMS

Central Reservation System (CRS): Centralized control of the booking of hotels rooms. The major features include: instant booking and confirmation, on line room inventory status, booking acceptance for events and venues, arrival list generation, multiple type of revenue reports, customer profile, comprehensive tariff management, and performance auditing.

CLICK/Valued Member: A marketing program targeted at frequent customers using a restaurants' credit card transactions. It identifies, tracks, rewards, and communicates with frequent diners and also sends the diners postcards to keep them abreast of special events, offers or menu changes.

Customer Relationship Management (CRM): A managerial philosophy that enables a firm to become intimately familiar with its customers. CRM focuses on maximizing revenue from each customer over the lifetime of the relationship by getting to know each one intimately.

Database Marketing: Extensive analysis of guest history, reservation databases, guest accounting/history, guest comment cards, brochure requests, Internet and other sources of guest data.

Data Mining in the Hospitality Industry: A largely automated process that uses statistical analyses to sift through massive data sets to detect useful, non-obvious, and previously unknown patterns or data trends for the hospitality industry. It emphasizes the computer-based exploration of previously uncharted relationship.

Data Warehouse: A corporate-wide database-management system that allows a company to manipulate large volumes of data in ways that are useful to the company: establishing sources, cleaning, organizing, summarizing, describing, and storing large amounts of data to be transformed, analyzed, and reported.

On-Line Analytical Processing (OLAP) Engines: Systems that generate query-based correlations in line with use-defined parameters and that make restaurants compare sales relative to plan by quarter and region.

Point-of-Sales (POS): Efficient data communications and affordable data storage using computers or specialized terminals that are combined with cash registers, bas code readers, optical scanners and magnetic stripe readers for accurately and instantly capturing the transaction.

Property Management System (PMS): Computerized system which deals with guest bookings, online reservations, point of sale, telephone and other amenities. Some property management systems also include payroll, back office and accounts.

Data Mining for Fraud Detection System

Roberto Marmo
University of Pavia, Italy

INTRODUCTION

As a conseguence of expansion of modern technology, the number and scenario of fraud are increasing dramatically. Therefore, the reputation blemish and losses caused are primary motivations for technologies and methodologies for fraud detection that have been applied successfully in some economic activities. The detection involves monitoring the behavior of users based on huge data sets such as the logged data and user behavior.

The aim of this contribution is to show some data mining techniques for fraud detection and prevention with applications in credit card and telecommunications, within a business of mining the data to achieve higher cost savings, and also in the interests of determining potential legal evidence.

The problem is very difficult because fraudsters takes many different forms and are adaptive, so they will usually look for ways to avoid every security measures.

BACKGROUND

The economic operations under security control can be classified into the class of genuine and into the class of fraudulent. A fraud is a criminal deception, use of false representations to obtain an unjust advantage, or to injure the rights and interests of another. The fraud is prevalent in insurance, credit card, telecommunications, health care, finance, etc. Diversity of fraud regards organisations, governments, and individuals such as external parties, internal employees, customers, service providers and suppliers.

It is important to analyze in detail the fraud scenario in order to establish: what is the fraudulent and normal behavior and what separates one individual from another, the degree of available knowledge about known fraud, the kind of available data exemplifying, types of fraud offenders and their modus operandi over time. It is difficult to provide precise estimates since some fraud may never be detected, and the operators are reluctant to reveal fraud losses due to show an appearance of reliability and security in business operations and to avoid reputation blemish.

It is necessary to take into account the cost of the fraud detection and the cost of fraudulent behavior, because stopping a fraud of few dollars can require a very expensive system. This is possible by introducing a decision layer on top of the system in order to decide the action taking into account factors like the amount of transaction and the risk associated to user doing the transaction.

The development of new detections methods is more difficult due to the severe limitation on privacy and on exchange of ideas. Moreover, data sets are not available and results are often not disclosed to the public.

The planning audit strategies is a posteriori fraud detection problem with prevention purpose of analyzing historical audit data and constructing models of planning effectively future audits. An application is fiscal and insurance domain, where audits are intended to detect tax evasion and fraudulent claims. A case study is presented by Bonchi (1999) which illustrates how techniques based on classification can be used to support the task of planning audit strategies.

The fraud detection methods in online auction (Shah, 2002) are based on statistical methods and association analysis in order to detect shilling, that occurs when the seller tries to hike up the prices in auction by placing buy bids under distinct aliases or through associates.

Apart fraud, the detection efforts may be further motivated by the need to understand the behavior of customers to enable provision of matching services and to improve operations.

DATA MINING APPROACH

Data mining analyzes the huge volumes of transactions and billing data and seeks out patterns, trends and clusters that reveal fraud. The main steps for implementing this approach for fraud detection within a business organization are:

1. Analyze the fraud objectives and the potential fraudsters, in order to converting them into data mining objectives;
2. Data collection and understanding;
3. Data cleaning and preparation for the algorithms;
4. Experiment design;
5. Evaluation results in order to review the process.

Relevant technical problems are due to:

1. Imperfect data not collected for purpose of data mining, so they are inaccurate, incomplete, and irrelevant data attributes;
2. Highly skewed data, there are many more legitimate than fraudulent examples, so by predicting all examples to be legal a very high success rate is achieved without detecting any fraud;
3. Higher chances of overfitting, that occurs when model high accuracy arises from fitting patterns in the training set that are not statistically reliable and not available in the score set.

To handle with skewed data the training set is divided into pieces where the distribution is less skewed (Chan, 1998).

A typical detection approach consists in outlier detection where the non-fraudulent behavior is assumed as normal and identify outliers that fall far outside the expected range should be evaluated more closely. Statistic techniques used for this approach are:

1. Predict and Classify
   - Regression algorithms: neural networks, CART, Regression, GLM;
   - Classification algorithms (predict symbolic outcome): CART, logistic regression;
2. Group and Find Associations
   - Clustering/Grouping algorithms: K-means, Kohonen, Factor analysis;
   - Association algorithms: GRI, Capri Sequence.

Many existing fraud detection systems operate by: supervised approaches on labelled data, hybrid approaches on labelled data, semi-supervised approaches with legal (non-fraud) data, unsupervised approaches with unlabelled data (Phua, 2005).

For a pattern recognition problem requiring great flexibility, adaptivity, and speed, neural networks techniques are suitable and the computational immunological systems, inspired by human immune system, might prove even more effective than neural networks in rooting out e-commerce fraud. (Weatherford, 2002). Unsupervised neural networks can mainly be used because they act on unlabelled data in order to extract an efficient internal representation of the data distribution structure.

The relational approach (Kovalerchuk, 2000) is applicables for discovering financial fraud, because overcome difficulties of traditional data mining in discovering patterns having only few relevant events in irrelevant data and insufficient statistics of relevant data.

The choice of three classification algorithms and one hybrid meta-learning was introduced by Phua (2004), to process the sampled data partitions, combined with straightforward cost model to evaluate the classifiers, so the best mix of classifiers can be picked.

Visualization techniques are based on the human capacity in pattern recognition in order to detect anomalies and are provided with real-time data. A machine-based detection method is static, the human visual system is dynamic and can easily adapt to the typical ever-changing techniques of the fraudsters. Visual data mining is a data mining approach that combine human detection and statistical analysis for greater computational capacity, is developed by building a user interface to manipulate the visual representation of data in fraud analysis.

Service providers use performance metric like the detection rate, false alarm rate, average time to detection after fraud starts, and average number of fraud or minutes until detection. As first step it is important to define a specific metric considering that misclassification costs can differ in each data set and can change over time. The false alarm rate is the percentage of legitimate that are incorrectly identified as fraudulent; fraud catching rate (or true positive rate or detection accuracy rate) is the percentage of transactions that are correctly identified as fraudulent; false negative rate is the percentage of transactions that are incorrectly identified as legitimate. The objective of this detection is to maximize correct fraud predictions and maintain incorrect predictions at an acceptable level. A realistic objective consists on balance of the performance criteria. A false negative error is usually more costly than

a false positive error. Other important considerations are: how fast the frauds can be detected and average time of detection, how many kind of fraud detected, on-line or off-line detection mode.

Five data mining tools was compared in Abbott (1998) with descriptions of their distinctive strenghts and weaknesses during the process of evaluating the tools. A list of software solutions is available on www.kdnuggets.com/solutions/fraud-detection.html

Credit Card Fraud

Improved fraud detection is an essential step to maintain the security and the suitability of the electronic payment system. There are two class: a stolen credit card number is used for making a purchase that appears on account statement, the cardholders used a legitimate credit card but deliberately denies the transaction upon receiving the service because a dishonest merchant use the card without permission from the cardholders (Leung, 2004).

Offline fraud is committed by using a stolen physical card and the institution issuing the card can lock it before it is used in a fraud. Online fraud is committed via web or phone shopping because only the card‘s details are needed and a manual signature and card imprint are not required (Lu).

Dorronsoro (1997) describe two characteristics: a very limited time span for decisions and huge amount of credit card operations to be processed that continue to grow in number.

Scalable techniques to analyze massive amounts of transaction data that efficiently compute fraud detectors in a timely manner is an important problem, especially for e-commerce (Chan, 1999).

Example of behaviour feature: value and kind of purchases in a specific time, number of charges made from specific postal codes. Behaviour analysis can also help uncover suspicious behaviors, such as a small gasoline purchase followed by an expensive purchase may indicate that someone was verifying that a charge card still worked before going after serious money. Departures from these norms may indicate fraud, expecially in case of two or three such instances ought to attract investigation.

Machine learning techniques such as backpropagation neural networks and Bayesian networks are compared for reasoning under uncertainty (Maes, 2002) showing that Bayesian networks are more accurate and much faster to train but they are slower when applied to new instances.

E-payment systems with internal logics that are capable of detecting fraud transaction are more powerful when combined with standard network security schemes, so there is a well user acceptance of credit card as payment solution. Leung (2004) designing issues on add-on module for detection in e-payment system, based on atomic transactions implemented in e-wallet accounts.

Telecommunications Fraud

The fraud in telecommunications, as any transmission of voice or data across a communications network, regards the intent of frauder to avoid or reduce legitimate call charges, so obtaining unbillable services, acquire free access, reselling the services. Significant loss are in wireless and international calls, because they are most expensive, so when fraud is committed the costs mount quickly. In this kind of application there are specific problems: callers are dissimilar, so calls that look like fraud for one account look like expected behavior for another, while all needles look the same. Sometime fraudsters create a new account without having the intention to pay for the used services. Moreover, fraud has to be found repeatedly, as fast as fraud calls are placed.

Due to legislation on privacy, the intentions of the mobile phone subscribers cannot be observed, but it is assumed that the intentions are reflected in the calling behavior and thus in the observed call data. The call data is subsequently used in describing behavioral patterns of users. Unfortunately, there is no specific sequence of communications that would be fraudulent with absolute certainty, because the same sequence of calls could as well be fraudulent or normal. Therefore, uncertainty in creating the model is needed. The calling activity of each mobile phone subscribers is recorded for billing in call records, which store attributes of calls like the identity of the subscriber, the number to call, time of the call, duration of the call, etc. Call data (or events input) are ordered in time, so it is possible to describe the daily usage formed by number and summed length of calls, and it is possible to separate them in sub-categories.

A common approach is to reduce the call records for an account to several statistics that are computed each period. For example, average call duration, longest call

duration, and numbers of calls to particular countries might be computed over the past hour, several hours, day, or several days. Account summaries can be compared to thresholds each period, and an account whose summary exceeds a threshold can be queued to be analyzed for fraud. The summaries that are monitored for fraud may be defined by subject matter experts,

It is possible to improve the thresholding (Fawcett, 1996) using an innovative method for choosing account-specific thresholds rather than universal thresholds that apply to all accounts. This approach detection has several disadvantages: thresholds may need to vary with time of day, type of account, and type of call to be sensitive to fraud without setting off too many false alarms for legitimate traffic, so it is necessary periodically to review the setting by an expert to accommodate changing traffic patterns. Moreover, accounts with high calling rates or unusual, but legitimate, calling patterns may setting off more false alarms. These problems lead to expensive multiplying the number of thresholds.

An efficient solution must be tailored to each account's own activity, based on historic memory of the use, and must be able to learn the pattern on an account and adapt to legitimate changes in user behavior. It is possible to define the user signature, in order to track legitimate behavior for an user, based on variables and values described by a multivariate probability distribution. The disadvantage consists in estimating the full multivariate distribution.

Cahill (2002) describes an approach based on tracking calling behavior on an account over time and scoring calls according to the extent that they deviate from that pattern and resemble fraud; signatures avoid the discontinuities inherent in most threshold-based systems that ignore calls earlier than the current time period.

The cloning fraud is a superimposition fraud upon the legitimate usage of an account. Fawcett (1997) describes a framework for automatically generating detectors for this cellular fraud, that is based on analysis of massive amounts of call data to determine patterns for a set of monitors watching customers' behavior with respect to one discovered pattern. A monitor profiles each customer behavior and measures the extent to which current behavior is abnormal with respect to the monitor particular pattern. Each monitor output is provided to a neural network, which weights the values and issues an alarm when the combined evidence for fraud is strong enough.

Ferreira (2006) describes the problem of superimposed fraud, that is the fraudsters make an illegitimate use of a legitimate account and some abnormal usage is blurred into the characteristic usage of the account. The approach is based on signature that corresponds to a set of information, or vector of feature variables, that captures the typical behavior of legitimate user during a certain period of time. In case of an user deviates from typical signature a statistical distance is evaluated to detect anomalies and to send an alarm.

Concerning visualization methods, it is possible to create a graphical representation of quantities of calls between different subscribers in various geographical locations, in order to detect international calling fraud (Cox, 1997). As example, in a simple representation the nodes correspond to the subscribers and the lines from the nodes to the countries encode the total number of calls, so investigations on the brightly colored nodes revealed that some of them were involved in fraud. In this way it is possible to show a complete set of information and several tools on a dashboard to facilitate the evaluation process.

FUTURE TRENDS

The available statistical models are unable to detect new types of fraud as they occur, because they must be aware of fraud methods such making a prevention effort, based on amounts of historical data, that can be implemented before the fraud occurred producing. Unfortunately, preventing fraud is an evolving and dynamic process and frauders creates new schemes daily, thus making not suitable this static solution. Future research is based on dynamic, self-correcting and self-adaptation in order to maintain continuous accuracy and adapt to identify and predict new scenarios. Another challenge is creating systems that work quickly enough to detect fraudulent activities as they occur. Moreover an useful feature is due to detailed results including the reason for suspicion of fraud, so the user can prioritizes the suspect transaction and related remedy.

CONCLUSION

The best strategy consists in: monitoring the behavior of populations of users, analyze contextual information and to avoid the examine of individual transactions

exclusively, good database with a large amount data on fraud behaviour, to combine the results from individual flags to create a score which raises a stop flag when it exceeds a threshold. Data mining based on machine learning techniques is suitable on economic data, in case of this approach is applied on a specific problem with a-priori analysis of the problem domain and a specific choice of performance metric. The papers of Bolton (2002) and Kou (2004) and the book of Wells (2004) present a survey of different techniques and more details to detect frauds in credit card, telecommunication, computer intrusion.

This work has been developed with the financial support of MIUR-PRIN funding "statistical models for e-business applications".

REFERENCES

Abbott, D. W., Matkovsky, I. P., & Elder, J. F. (1998). An evaluation of high-end data mining tools for fraud detection. In *Proceedings of the 1998 IEEE International Conference on Systems, Man, and Cybernetics*, vol. 3, 2836-2841.

Bolton, R. J., & Hand, D. J. (2002). Statistical fraud detection: a review. *Statistical Science,* 17(3), 235-255.

Bonchi, F., Giannotti, F., Mainetto, G., & Pedreschi, D. (1999). A classification-based methodology for planning audit strategies in fraud detection. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* 175-184.

Cahill M., Chen, F., Lambert, D., Pinheiro, J., & Sun, D. X. (2002). Detecting Fraud in the Real World. In: *Handbook of Massive Datasets*, Kluewer, 1-18.

Chan, P. K., Stolfo, S. J. (1998). Toward scalable learning with non-uniform class and cost distributions: a case study in credit card fraud detection. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, 164-168.

Chan, P.K., Fan, W., Prodromidis, A.L., & Stolfo, S.J. (1999). Distributed data mining in credit card fraud detection, *IEEE Intelligent Systems*, 14(6), 67-74.

Cox, K. C., Eick, S. G., Wills, G. J., & Brachman, R. J. (1997). Visual data mining: recognizing telephone calling fraud. *Journal of Data Mining and Knowledge Discover*, 1(2):225-231.

Fawcett, T. & Provost, F. (1996). Combining data mining and machine learning for effective user profiling, *Second International Conference on Knowledge Discovery and Data Mining*, 8-13.

Fawcett, T. & Provost, F. (1997). Adaptive fraud detection. D*ata mining and knowledge discovery*, 1(3), 291–316.

Ferreira, P., Alves, R., Belo, O., Cortesao, L. (2006). Establishing fraud detection patterns based on signatures. In Proceedings of ICDM, Springer Verlag, 526-538.

Kuo, Y.F., Lu, C.-T. Sirwongwattana, S., & Huang, Y.-P. (2004). Survey of fraud detection techniques. In *Proceedings of IEEE International Conference on Networking, Sensing & Control*, 749-753.

Kovalerchuk, B., & Vityaev, E. (2000). Data Mining in finance: advances in relational and hybrid methods, Kluwer Academic. Publisher, Boston.

Leung, A., Yan, Z., & Fong, S. (2004). On designing a flexible e-payment system with fraud detection capability. In *IEEE International Conference on E-Commerce Technology*, 236-243.

Maes, S., Tuyls, K., Vanschoenwinkel, B. & Manderick, B. (2002). Credit card fraud detection using bayesian and neural networks. In *Proceedings of International NAISO Congress on Neuro Fuzzy Technologies*.

Phua, C., Alahakoon, D., & Lee, V. (2004). Minority report in fraud detection: classification of skewed data. *ACM SIGKDD Explorations*, 6(1), 50-59.

Phua, C, Lee, V., Smith, K., & Gayler, R. (2005). A comprehensive survey of data mining-based fraud detection research, draft submitted from www.bsys.monash.edu.au/people/cphua/papers

Shah, H.S., Joshi, N.J., Wurman, P.R. (2002). Mining for bidding strategies on eBay, Springer, 2002.

Weatherford, M.(2002). Mining for fraud, IEEE Intelligent Systems, 3(7), 4-6.

Wells J.T. (2004), Corporate Fraud Handbook: Prevention and Detection, John Wiley & Sons.

KEY TERMS

Credit Card Fraud: A stolen credit card number is used for making a purchase, or a dishonest merchant use the card without permission from the cardholders.

Fraud: A fraud is a criminal deception, use of false representations to obtain an unjust advantage, or to injure the rights and interests of another.

Fraud Detection: An automatic system based on description of user behaviour and fraud scenario, in order to detect a fraudulent activity as soon as possible to reduce loss.

Telecommunications Fraud: any transmission of voice or data across a communications network to avoid or reduce legitimate call charges.

Data Mining for Improving Manufacturing Processes

Lior Rokach
Ben-Gurion University, Israel

INTRODUCTION

In many modern manufacturing plants, data that characterize the manufacturing process are electronically collected and stored in the organization's databases. Thus, data mining tools can be used for automatically discovering interesting and useful patterns in the manufacturing processes. These patterns can be subsequently exploited to enhance the whole manufacturing process in such areas as defect prevention and detection, reducing flow-time, increasing safety, etc.

When data mining is directed towards improving manufacturing process, there are certain distinctions that should be noted compared to the classical methods employed in quality engineering, such as the experimental design. In data mining the primary purpose of the targeted database is not data analysis; the volume of the collected data makes it impractical to explore it using standard statistical procedures (Braha and Shmilovici, 2003).

BACKGROUND

This chapter focuses on mining performance-related data in manufacturing. The performance can be measured in many different ways, most commonly as a quality measure. A product is considered as faulty when it does not meet its specifications. Faults may come from sources such as, raw material, machines setup and many other sources.

The quality measure can either have nominal values (such as "good"/"bad") or continuously numeric values (Such as the number of good chips obtained from silicon wafer or the pH level in a cream cheese). Even if the measure is numeric, it can still be reduced to a sufficiently discrete set of interesting ranges. Thus we can use classification methods in order to find the relation between the quality measure (target attribute) and the input attributes (the manufacturing process data).

Classification methods can be used to improve the learning curve both in the learning pace, as well as in the target measure that is reached at the mature stage. The idea is to find a classifier that is capable of predicting the measure value of a certain product or batch, based on its manufacturing parameters. Subsequently, the classifier can be used to set up the most appropriate parameters or to identify the reasons for bad measures values.

The manufacturing parameters obviously include the characteristics of the production line (such as which machine has been used in each step, how each machine has been setup, operation sequence etc.), and other parameters (if available) relating to the raw material that is used in the process; the environment (moistness, temperature, etc); the human resources that operate the production line (the experience level of the worker which have been assigned on each machine in the line, the shift number) and other such significant factors.

The performance measure (target attribute) in manufacturing data tends to have imbalanced distribution. For instance, if the quality measure is examined, then most of the batches pass the quality assurance examinations and only a few are considered invalid. On the other hand, the quality engineer is more interested in identifying the invalid cases (the less frequent class).

Traditionally, the objective of the classification method is to minimize the misclassification rate. However, for the unbalanced class distribution, accuracy is not an appropriate metric. A classifier working on a population where one class ("bad") represents only 1% of the examples can achieve a significantly high accuracy of 99% by just predicting all the examples to be of the prevalent class ("good"). Thus, the goal is to identify as many examples of the "bad" class as possible (high recall) with as little false alarms (high precision). Traditional methods fail to obtain high values of recall and precision for the less frequent classes, as they are oriented toward finding global high accuracy.

Figure 1. Decision tree for quality assurance

```
CW_WASH_DUR <= 286
|  FINAL_COOLING_TEMP <= 5.9
|  |  AVG_COOLING_TEMP <= 10.1: Tasty (0.864,0.136)
|  |  AVG_COOLING_TEMP > 10.1: Sour (0.323,0.674)
|  FINAL_COOLING_TEMP > 5.9
|  |  AVG_COOLING_TEMP <= 12.3: Tasty (0.682,0.318)
|  |  AVG_COOLING_TEMP > 12.3: Sour (0.286,0.714)
CW_WASH_DUR > 286: Tasty (0.906,0.094)
```

Usually in manufacturing plants there are many input attributes that may affect performance measure and the required number of labelled instances for supervised classification increases as a function of dimensionality. In quality engineering mining problems, we would like to understand the quality patterns as soon as possible in order improve the learning curve. Thus, the training set is usually too small relative to the number of input features.

MAIN FOCUS

Classification Methods

Classification methods are frequently used in mining of manufacturing datasets. Classifiers can be used to control the manufacturing process in order to deliver high quality products.

Kusiak (2002) suggested a meta-controller seamlessly developed using neural network in order to control the manufacturing in the metal industries. While neural networks provide high accuracy, it is usually hard to interpret its predictions. Rokach and Maimon (2006) used a decision tree inducer in cheese manufacturing. As in every dairy product, there is a chance that a specific batch will be found sour when consumed by the customer, prior to the end of the product's shelf-life. During its shelf-life, the product's pH value normally drops. When it reaches a certain value, the consumer reacts to it as a spoiled product. The dairy department performs randomly tests for pH as well organoleptic (taste) at the end of the shelf-life.

Figure 1 demonstrates a typical decision tree induced from the manufacturing database. Each representing symptoms of a product quality, denoted here as features Cold Water Wash Duration, Final Cooling Temperature and Average Cooling Temperature. The leaves are labeled with the most frequent class together with their appropriate probability. For instance, the probability to get a tasty cheese is 0.906 if the Cold Water Wash Duration is greater than 286 seconds.

Ensemble of Classifiers

The main idea of ensemble methodology is to combine a set of models, each of which solves the same original task, in order to obtain a better composite global model, with more accurate estimates than can be obtained from using a single model.

Maimon and Rokach (2004) showed that ensemble of decision trees can be used in manufacturing datasets and significantly improve the accuracy. Similar results have been obtained by Braha and Shmilovici (2002) arguing that ensemble methodology is particularly important for semiconductor manufacturing environments where various physical and chemical parameters that affect the process exhibit highly complex interactions, and data is scarce and costly for emerging new technologies.

Because in many classification applications non-uniform misclassification costs are the governing rule, has led to a resurgence of interest in cost-sensitive classification. Braha et al. (in press) present a decision-theoretic classification framework that is based on a model for evaluating classifiers in terms of their value

in decision-making. The underlying assumption is that the predictions displayed by a classifier are 'effective' only insofar as the derived information leads to actions that increase the payoff for the decision-maker. They suggested two robust ensemble classification methods that construct composite classifiers which are at least as good as any of the existing component classifiers for all possible payoff functions and class distributions.

Decomposition Methods

When a problem becomes more complex, there is a natural tendency to try to break it down into smaller, distinct but connected pieces. This approach is generally referred to as decomposition. Several researchers have shown that the decomposition methodology can be appropriate for mining manufacturing data (Kusiak, 2000; Maimon and Rokach, 2001). Recently Rokach and Maimon (2006) suggested a new feature set decomposition algorithm called Breadth-Oblivious-Wrapper for manufacturing problems. Feature set decomposition decomposes the original set of features into several subsets, and builds a classifier for each subset. A set of classifiers are trained such that each classifier employs a different subset of the original features set. An unlabeled instance is classified by combining the classifications of all classifiers. This approach can be further improved by using genetic algorithms (Rokach, in press).

Association Rules Methods

Agard and Kusiak (2004) use data mining for selection of subassemblies in wire harness process that the automotive industry. By using association rules methods they construct a model for selection of subassemblies for timely delivery from the suppliers to the contractor. The proposed knowledge discovery and optimization framework integrates the concepts from product design and manufacturing efficiency.

da Cunha et al. (2006) analyzed the effect of the production sequence on the quality of the product using data mining techniques. The sequence analysis is particularly important for assemble-to-order production strategy. This strategy is useful when the basic components can be used as building blocks of a variety of end products. Manufactures are obligated to manage a growing product portfolio in order to meet customer needs. It is not feasible for the manufacturer to build all possible end products in advance, thus the assembly is performed only when the customer order is received. Because the quality tests of the components can be performed in advance, the quality of the final assembly operations should be the focus. da Cunha et al., (2006) provide an industrial example of electrical wire harnesses in the automobile industry. There are millions of different wire harnesses for a unique car model, mainly because wire harnesses control many functions (such as the electrical windows) and because every function has different versions depending on other parameters such as engine type.

In order to solve the problem using existing techniques the following encoding is proposed: Each column in the table represents a different sequence pair. By that "the precedence information is contained in the data itself". For instance Figure 2 illustrates a product route which consists of 5 operations followed by an inspection test targeted to identify faults. Because a fault was identified, rework of operation 9 has to be done. The rework is followed by an additional inspection test that indicates that the product fits the product specification.

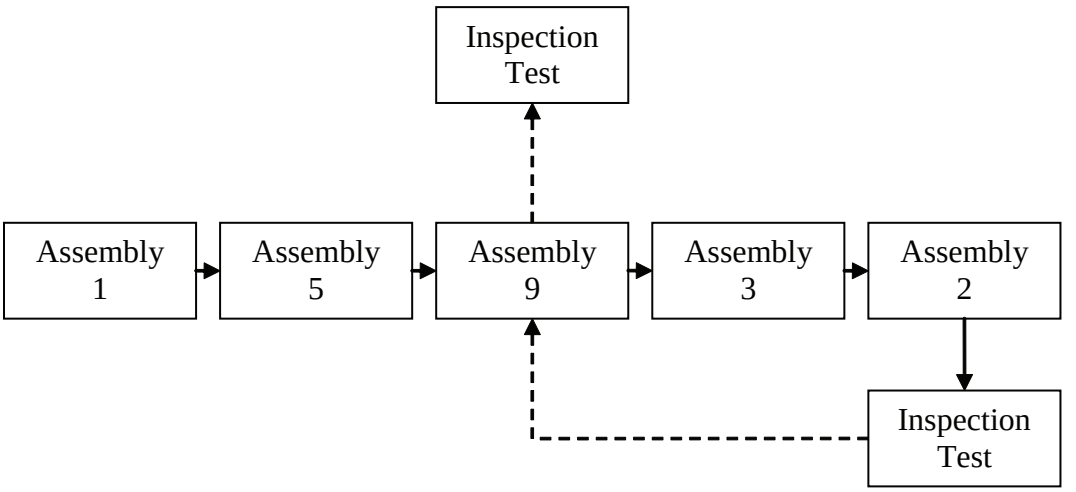
da Cunha et al., (2006) suggest to represent the sequence 1-5-9-3-2 by the following set of pairs: {B1, 1_5 , 5_9, 9_3, 3_2, 2F}. Note that B1 and 2F represent the fact that the sequence begin in operation 1 and that the final operation is 2. By proposing this encoding, it is possible to analyze the effect of the production sequence on the quality of the product using supervised association rules algorithms.

Applications in the Electronic Industry

Most of the data mining applications in manufacturing have been applied to the electronic industry. The reason for that is four-fold:

1. **Process Complexity:** The manufacturing process (such as semiconductor wafer fabrication) = contains many steps. It is very difficult to use standard statistical procedures to detect intricate relations between the parameters of different processing steps (Braha and Shmilovici, 2003).
2. **Process variations:** The process suffers from constant variations in the characteristics of the wafer manufacturing process (Braha et al., in press). It has been estimated that up to 80% of yield losses in high volume integrated circuits production can be attributed to random, equipment

Figure 2. Illustration of a product route



related process-induced defects. The manufacturing environments at different wafer producing plants may be similar, still some wafers obtain higher quality than other. Even within individual plants product quality varies (Kusiak, 2006).

3. **Data availability:** In leading fabs, large volume of data is collected from the production floor.
4. **Data Volume:** Semiconductor databases are characterized as having large amount of records, each of which contain hundreds of attributes, which need to be simultaneously considered in order to accurately model the system's behavior (Braha and Shmilovici, 2003, Rokach and Maimon, 2006).

The main performance measures that are often used to assess profitability in the semiconductor manufacturing process are yield and flow times of production batches (Cunningham et al 1995). Yield models are developed to automate the identification of out of control manufacturing conditions. This is used as feedback to improve and confirm top quality operation.

Last and Kandel (2001) applied the Info-Fuzzy Network (IFN) methodology of data mining and knowledge discovery to WIP (Work-in-Process) data obtained from a semiconductor plant. In this methodology, the recorded features of each manufacturing batch include its design parameters, process tracking data, line yield, etc.

Electronic product assembly lines face a quality problem with printed-circuit boards where assemblies rather than components fail the quality test (Kusiak, 2006). Classification methods can be used to predict circumstances under which an individual product might fail. Kusiak and Kurasek (2001) presents a an industrial case study in which data mining has been applied to solve a quality engineering problem in electronics assembly. During the assembly process, solder balls occur underneath some components of printed circuit boards. The goal is to identify the cause of solder defects in a circuit board using a rough set approach over the decision tree approach.

Braha and Shmilovici (2003) presented an application of decision tree induction to lithographic process. The work suggests that decision tree induction may be particularly useful when data is multidimensional, and the various process parameters and machinery exhibit highly complex interactions. Wu et al. (2004) suggested an intelligent CIM (computer integrated manufacturing) system which uses decision tree induction algorithm. The proposed system has been implemented in a semiconductor packaging factory and an improved yield is reported. Jothishankar et al. (2004) used data mining techniques for improving the soldering process of printed circuit boards. More specifically they looked for the reasons for a specific defect in which a chip component that has partially or completely lifted off one end of the surface of the pad.

Fountain et al. (2003) describe a data mining application to the problem of die-level functional tests (DLFT) in integrated circuit manufacturing. They

describe a decision-theoretic approach to DLFT in which historical test data is mined to create a probabilistic model of patterns of die failure. This model is combined with greedy value-of-information computations to decide in real time which die to test next and when to stop testing.

Braha and Shmilovici (2002) presented an application of refinement of a new dry cleaning technology that utilizes a laser beam for the removal of micro-contaminants which have a negative impact on the process yield. They examine two classification methods (decision tree induction and neural networks) showing that both of them can significantly improve the process. Gardner and Bieker (2000) suggested a combination of self-organizing neural networks and rule induction in order to identify the critical poor yield factors from normally collected wafer manufacturing data from Motorola manufacturing plant, showing that the proposed method efficiently identify lower yielding factor and in a much quicker way. Kim and Lee (1997) used explicit and implicit methods to predict nonlinear chaotic behavior of manufacturing process. They combined statistical procedures, neural networks and case based reasoning for prediction of manufacturing processed of optic fibers adulterated by various patterns of noise.

FUTURE TRENDS

There are several trends in the field of using data mining in manufacturing systems including:

- Preliminary applications have used existing data mining techniques to solve manufacturing problems. However data accumulated in manufacturing plants have unique characteristics, such as unbalanced distribution of the target attribute, and a small training set relative to the number of input features. Thus, data mining methods should be adapted accordantly.
- It seems that it is important to integrate the mining of manufacturing process with other entities such as equipment failure events.
- The creation of a unified knowledge discovery framework for manufacturing process.
- Developing a meta-data mining approach, in which knowledge about the organization will be used to improve the effectiveness of data mining methods.
- Developing a distributed data mining framework by sharing patterns among organizations while preserving privacy.

CONCLUSION

In this chapter, we surveyed data mining methods and their applications in intelligent manufacturing systems. Analysis of past performance of production systems is necessary in any manufacturing plan in order to improve manufacturing quality or throughput. The difficulty is in finding pertinent information as in the manufacturing databases. Data mining tools can be very beneficial for discovering interesting and useful patterns in complicated manufacturing processes. These patterns can be used, for example, to improve quality. We review several applications in the field especially in the electronic industry. We also analyzed how ensemble methods and decomposition methods can be used to improve prediction accuracy in this case. Moreover we show how association rules can be used in manufacturing applications. Finally future trends in this field have been presented.

REFERENCES

Agard B., and Kusiak A., Data Mining for Subassembly Selection, ASME Transactions: Journal of Manufacturing Science and Engineering, 126(3), 627-631.

Agard, B. and Kusiak, A., A data-mining based methodology for the design of product families. Int. J. Prod. Res., 2004, 42(15), 2955–2969.

Braha D., Elovici Y., & Last M. (in press), A Theory of Actionable Data Mining with Application to Semiconductor Manufacturing Control, International Journal of Production Research.

Braha, D., & Shmilovici, A. (2002) Data Mining for Improving A Cleaning Process in the Semiconductor Industry. *IEEE Transactions on Semiconductor Manufacturing,* 15(1), 91-101.

Braha, D., & Shmilovici, A., "On the use of Decision Tree Induction for Discovery of Interactions in a Photolithographic Process," IEEE Transactions on

Semiconductor Manufacturing, vol. 16 (4), pp. 644-652, 2003.

Cunningham, S. P., Spanos C. J., & Voros, K., "Semiconductor yield improvement: Results and best practices," IEEE Trans. Semiconduct. Manufact., vol. 8, pp. 103–109, May 1995.

da Cunha C. , Agard B., & A. Kusiak, Data mining for improvement of product quality, International Journal of Production Research, Volume 44, Numbers 18-19, -19/15 September-1 October 2006, pp. 4027-4041.

Fountain T., Dietterich T., & Sudyka Bill (2003), Data mining for manufacturing control: an application in optimizing IC tests, Exploring artificial intelligence in the new millennium, Morgan Kaufmann Publishers Inc , pp. 381 – 400 .

Gardner M., & Bieker J. (2000), Data mining solves tough semiconductor manufacturing problems, Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, Boston, Massachusetts, United States, pp. 376 – 383.

Jothishankar, M. C., Wu, T., Johnie, R. & Shiau, J-Y., "Case Study: Applying Data Mining to Defect Diagnosis", Journal of Advanced Manufacturing Systems, Vol. 3, No. 1, pp. 69-83, 2004.

Kim S.H., & Lee C.M., Nonlinear prediction of manufacturing systems through explicit and implicit data mining (1997) Computers and Industrial Engineering, 33 (3-4), pp. 461-464.

Kusiak A., and Kurasek C., Data Mining of Printed-Circuit Board Defects, IEEE Transactions on Robotics and Automation, Vol. 17, No. 2, 2001, pp. 191-196

Kusiak A., Rough Set Theory: A Data Mining Tool for Semiconductor Manufacturing, IEEE Transactions on Electronics Packaging Manufacturing, Vol. 24, No. 1, 2001, pp. 44-50.

Kusiak, A., A data mining approach for generation of control signatures. ASME Trans.: J. Manuf. Sci. Eng., 2002, 124(4), 923–926.

Kusiak A., Data Mining: Manufacturing and Service Applications, International Journal of Production Research, Vol. 44, Nos 18/19, 2006, pp. 4175-4191.

Last, M., & Kandel, A., "Data Mining for Process and Quality Control in the Semiconductor Industry", In Data Mining for Design and Manufacturing: Methods and Applications, D. Braha (ed.), Kluwer Academic Publishers, pp. 207-234, 2001.

Maimon O., & Rokach L. (2001), "Data Mining by Attribute Decomposition with Semiconductors Manufacturing Case Study", in Data Mining for Design and Manufacturing: Methods and Applications (Editor: D. Braha), Kluwer Academic Publishers, pp 311-336.

Maimon O., & Rokach L., "Ensemble of Decision Trees for Mining Manufacturing Data Sets", Machine Engineering, vol. 4 No1-2, 2004.

Rokach L., & Maimon O., "Data mining for improving the quality of manufacturing: a feature set decomposition approach", Journal of Intelligent Manufacturing, 17(3), 2006, pp. 285–299**.**

Rokach L. (in press), "Mining Manufacturing Data using Genetic Algorithm-Based Feature Set Decomposition", International Journal of Intelligent Systems Technologies and Applications.

Wu R., Chen R., & Fan C.R. (2004), Design an intelligent CIM system based on data mining technology for new manufacturing processes, International Journal of Materials and Product Technology - Vol. 21, No.6 pp. 487 – 504

KEY TERMS

Association Rules: Techniques that find in a database conjunctive implication rules of the form ``X and Y implies A and B''.

CIM (Computer Integrated Manufacturing): The complete automation of a manufacturing plant in which all processes are functioning under computer control.

Classifier: A structured model that maps unlabeled instances to finite set of classes.

Clustering: The process of grouping data instances into subsets in such a manner that similar instances are grouped together into the same cluster, while different instances belong to different clusters.

Ensemble of Classifiers: Combining a set of classifiers in order to obtain a better composite global

classifier, with more accurate and reliable estimates or decisions than can be obtained from using a single classifier.

Induction Algorithm: An algorithm that takes as input a certain set of instances and produces a model that generalizes these instances.

Intelligent Manufacturing Systems: A manufacturing system that aims to satisfy customer needs at the most efficient level for the lowest possible cost by using AI methods.

Quality Control: A set of measures taken to ensure that defective products or services are not produced, and that the design meets performance requirements.

Data Mining for Internationalization

Luciana Dalla Valle
University of Milan, Italy

INTRODUCTION

The term "internationalization" refers to the process of international expansion of firms realized through different mechanisms such as export, strategic alliances and foreign direct investments. The process of internationalization has recently received increasing attention mainly because it is at the very heart of the globalization phenomenon. Through internationalization firms strive to improve their profitability, coming across new opportunities but also facing new risks.

Research in this field mainly focuses on the determinants of a firms' performance, in order to identify the best entry mode for a foreign market, the most promising locations and the international factors that explain an international firms' performance. In this way, scholars try to identify the best combination of firms' resources and location in order to maximize profit and control for risks (for a review of the studies on the impact of internationalization on performance see Contractor et al., 2003).

The opportunity to use large databases on firms' international expansion has raised the interesting question concerning the main data mining tools that can be applied in order to define the best possible internationalization strategies.

The aim of this paper is to discuss the most important statistical techniques that have been implemented to show the relationship among firm performance and its determinants.

These methods belong to the family of multivariate statistical methods and can be grouped into *Regression Models* and *Causal Models*. The former are more common and easy to interpret, but they can only describe direct relationships among variables; the latter have been used less frequently, but their complexity allows us to identify important causal structures, that otherwise would be hidden.

BACKGROUND

We now describe the most basic approaches used for internationalization. Our aim is to give an overview of the statistical models that are most frequently applied in International Finance papers, for their easy implementation and the straightforwardness of their result interpretation. In this paragraph we also introduce the notation that will be used in the following sections.

In the class of *Regression Models*, the most common technique used to study internationalization is the *Multiple Linear Regression*. It is used to model the relationship between a continuous response and two or more linear predictors.

Suppose we wish to consider a database, with N observations, representing the enterprises, the response variable (firm performance) and the covariates (firm characteristics). If we name the dependent variable Y_i, with $i = 1, \ldots, N$, this is a linear function of H predictors $x_1, \ldots, x_H$, taking values $x_{i1}, \ldots, x_{iH}$, for the i–th unit. Therefore, we can express our model in the following way:

$$y_i = \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_H x_{iH} + \varepsilon_i, \qquad (1)$$

where $\beta_1, \ldots, \beta_i$ are the regression coefficients and ε_i is the error term, that we assume to be normally distributed with mean zero and variance σ^2 (Kutner et al., 2004).

Hitt, Hoskisson and Kim (1997), for example, applied the technique shown above to understand the influence of firm performance on international diversification, allowing not only for linear, but also for curvilinear and interaction effects for the covariates.

When our dependent variable is discrete, the most appropriate regression model is the *Logistic Regression* (Giudici, 2003).

The simplest case of Logistic Regression is when the response variable Y_i is binary, so that it assumes only the two values *0* and *1*, with probability p_i and $1\text{-}p_i$, respectively.

We can describe the Logistic Regression model for the *i*-th unit of interest by the logit of the probability p_i, linear function of the predictors:

$$\log it(p_i) = \log \frac{p_i}{1-p_i} = \eta_i = \underline{x}_i' \underline{\beta}, \qquad (2)$$

where $\underline{x}_i$ is the vector of covariates and $\underline{\beta}$ is the vector of regression coefficients.

For more details about Logit models the reader can consult, for example, Mardia, Kent and Bibby (1979) or Cramer (2003).

For an interesting application, the reader can consult the study by Beamish and Ruihua (2002), concerning the declining profitability from foreign direct investments by multinational enterprises in China. In their paper the authors used subsidiaries' performance as the response variable, with the value "1" indicating "profitable" and value "0" denoting "break-even" or "loss" firms. Beamish and Ruihua show that the most important determinants of profitability are subsidiaries-specific features, rather than macro-level factors.

Concerning *Causal models*, *Path Analysis* is one of the first techniques introduced in the internationalization field.

Path Analysis is an extension of the regression model, where causal links are allowed between the variables. Therefore, variables can be at the same time both dependent and independent (see Dillon and Goldstein, 1984 or Loehlin, 1998).

This approach could be useful in an analysis of enterprise internationalization, since, through a flexible covariate modeling, it shows even indirect links between the performance determinants.

Path Analysis distinguishes variables into two different types: *exogenous* and *endogenous*. The former are always independent and they do not have explicit causes; the latter can be dependent as well as independent and, since they are stochastic, they are characterized by a disturbance term as an uncertainty indicator.

The association between a couple of endogenous variables caused by an exogenous one is defined as *spurious correlation*. This particular correlation is depicted by an arrow in a circle-and-arrow figure, called a *path diagram* (see figure 1).

The object of Path Analysis is to find the best causal model through a comparison between the regression weights predicted by some models and the observed correlation matrix for the variables. Those regression weights are called *path coefficients* and they show the direct effect of an independent variable on a dependent variable in the path model.

For more details about Path Analysis, see, for example, Bollen (1989) or Deshpande and Zaltman (1982).

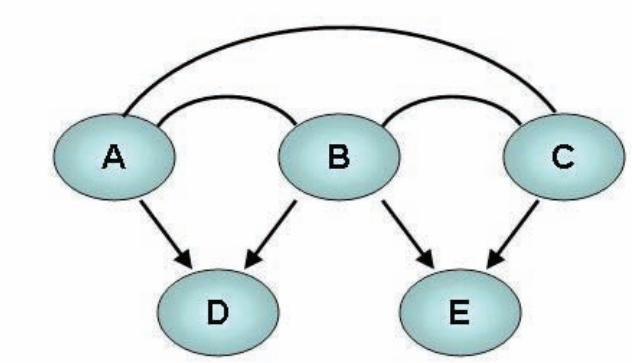
MAIN FOCUS

In this section we analyze the statistical methodologies most recently used for internationalization, with particular attention to the applications.

Regression Models

Logistic Regression is very useful when we deal with a binary response variable, but when we consider a

Figure 1: The path diagram. A, B and C are correlated exogenous variables; D and E are endogenous variables caused by A, B and C.



dependent variable with more than two response categories we need to consider a different type of model: the *Multinomial Logit Model* (see, for example, Hensher et al., 2005 or Mc Fadden, 1974).

Suppose to have a random variable Y_i, as the response variable, which takes values indexed by *1, 2, …, J*. Then,

$$p_{ij} = \Pr\{Y_i = j\} \qquad (3)$$

denotes the probability that the *i*-th response falls in the *j*-th category. Since the categories are mutually exclusive and exhaustive, we have only *J-1* parameters, because

$$\sum_{j=1}^{J} p_{ij} = 1. \qquad (4)$$

Typically, in the Multinomial Logit Model, the most common approach consists of nominating one of the categories of the response variable as the baseline (normally the last category), calculating the logit for all the other categories relative to the baseline and finally setting the logit as a linear function of the predictors:

$$\eta_{ij} = \log \frac{p_{ij}}{p_{iJ}} = \underline{x}_i \, ' \, \underline{\beta}_j, \qquad (5)$$

where $\underline{\beta}_j$ is the vector of regression coefficients, for *j = 1, …, J-1*.

If our aim is to understand the choice behavior of individuals, we introduce an extension of the Multinomial Logit model, where the explanatory variables may include attributes of the alternative behaviors as well as characteristics of the individuals: the *Conditional Logit Model*, proposed by Mc Fadden (1974).

Consider the random variable Y_i, representing the choice among *J* alternative behaviors. Suppose that U_{ij} represents the *i*-th individual's utility resulting from the *j*-th alternative, so that

$$U_{ij} = \eta_{ij} + \varepsilon_{ij}, \qquad (6)$$

where η_{ij} is the systematic component and ε_{ij} is the random component (for more details, see Greene (1997)). This disturbance term reflects the unique advantages of alternative *j* to individual *i*. It differs across alternatives for any one individual, and across individuals for any alternative.

Assuming that individuals act in a rational way, they tend to maximize their utility. Thus, the probability that subject *i* will choose alternative *j* is given by

$$p_{ij} = \Pr\{Y_i = j\} = \Pr\{U_{ij} > U_{ik}, k \neq j\}. \qquad (7)$$

It can be shown that if the random components ε_{ij} are distributed according to a Type I extreme value distribution, then we can express these probabilities as:

$$p_{ij} = \frac{\exp\{\eta_{ij}\}}{\sum_{k=1}^{J} \exp\{\eta_{ij}\}}. \qquad (8)$$

The previous equation can be obtained according to the *axiom of independence from irrelevant alternatives*, requiring that the odds of choosing alternative *j* over alternative *k* should be independent of the choice set for all pairs *j, k*. However, this requirement is not always reasonable for all the choices.

The main difference between the Conditional and the Multinomial Logit model is the fact that in the latter the expected utilities η_{ij} are a linear function of the individuals' characteristics. Instead, in the *Conditional Logit Model* the expected utilities are explained in terms of the alternative characteristics, rather than the individual attributes.

In International Business studies Conditional Logit models are applied when the main objective is to determine which factors are behind the multinational enterprises' location choices. Chang and Park (2005), for example, analyze a database of Korean firms investing in China. In their paper U_{ij} represents the utility a firm *i* derives from opening a manufacturing operation in region *j*. In particular, a firm decides to locate in region *j* if U_{ij} is the maximum among all the utilities. The model's response variable is expressed in terms of the random variable Y_i that indicates a choice made by firm *i*, assuming independence of irrelevant alternatives (the relative probability of choosing two alternatives does not depend on the availability of other alternatives).

The Conditional Logit model is also applied by Ford and Strange (1999) in their paper about Japanese firms locating affiliates in Western Europe. Each Japanese investor chooses to locate its affiliate *i* in country *j* if the expected profit (utility) is maximized. Since the random variable Y_i indicates the location chosen for

affiliate *i*, then the probability of choosing a specific country *j* depends upon the attributes of that country relative to the attributes of the other seven countries in the choice set.

Another interesting extension of the Logit model is the *Hierarchical Logit model,* which defines a hierarchy of nested comparisons between two subsets of responses, using a logit model for each comparison (see, for example, Louviere et al., 2000 or Train, 2003). This approach is particularly appropriate if we assume that individuals make decisions on a sequential basis.

An application of the Nested Logit Model is described in the paper by Basile, Castellani and Zanfei (2003). The database used is about 5761 foreign subsidiaries established in 55 regions in 8 EU countries. The authors' aim is to understand the determinants of multinational firms' location choices not only in European countries, but also in the regions within each country. The idea is to group alternatives (regions) into nests (countries). Let the *J* regions being grouped into *K* countries, then the probability of choosing region *j* is calculated as the product of two probabilities: the probability of choosing region *j* conditional on having chosen country *k* times the marginal probability of choosing country *k*. The final model expresses firm's utility in terms of country characteristics and of regional characteristics, which eventually vary across firms.

Causal Models

When the variables in the model are latent variables measured by multiple observed indicators, Path Analysis is termed *Structural Equation Modeling* (*SEM*). We can define such models according to Goldberger (1973): "by 'structural models' I refer to stochastic models in which each equation represents a causal link rather than an empirical association". Therefore this technique can be seen as an extension of Path Analysis (since each variable has not only a single but multiple indicators) and of Factor Analysis (since it permits direct effects connecting those variables).

Variables in SEM are grouped into observed and unobserved. The former are also called *indicators* and are used to measure the unobserved variables (*factors* or latent variables), that could be exogenous as well as endogenous (Mueller, 1996).

To determine the causal structure among the variables, first of all the null model (without direct effects connecting factors) is built. Then, this model is tested to evaluate the significance of the latent variables. Finally, more complex models are compared to the null model and are accepted or rejected, according to a goodness of fit measure. Once the final model is determined, the *direct structural standardized coefficients* are estimated via maximum likelihood estimation or generalized least squares. These coefficients are used to compare direct effects on an endogenous variable.

The factor *loadings* are the weights of the indicator variables on the respective factor and they can be used to interpret the meaning of the latent variable (Bollen, 1989).

The database analyzed by Goerzen and Beamish (2003) is about a sample of 580 large Japanese multinational enterprises. The authors' purpose is to discover the relationship between the endogenous latent variable representing firm's economic performance and the two independent exogenous latent variables "international asset dispersion (IAD)" and "country environment diversity (CED)", through the observed variables (firm-specific, country-specific and control variables). The endogenous variable is measured by three indicators: "Jensen's alpha", "Sharpe's measure" and "Market-to-book ratio". The exogenous latent variable IAD is measured by the "number of countries", the "asset dispersion entropy score" and the "number of foreign subsidiaries". Finally, the associated indicators of CED are the "global competitiveness entropy score", the "economic freedom entropy score", the "political constraint entropy score" and the "cultural diversity entropy score". The equations describing the model for each firm are:

$$\begin{cases} \eta = \gamma_1 \xi_1 + \gamma_2 \xi_2 + \gamma_3 \xi_1 \xi_2 + \varsigma \\ x_i = \lambda_i^x \xi_1 + \delta_i & i = 1,2,3 \\ x_i = \lambda_i^x \xi_2 + \delta_i & i = 4,5,6,7 \\ x_i = \lambda_i^x \xi_1 \xi_2 + \delta_i & i = 8,9,\ldots,19 \\ y_j = \lambda_j^y \eta + \varepsilon_j & j = 1,2,3 \end{cases}$$

where η denotes the economic performance, ξ_1 denotes the international asset dispersion, ξ_2 denotes the country environment diversity, ς is the error term and γ_1, γ_2, γ_3 are the parameters associated to the independent variables. The equation is nonlinear, for the presence of the interaction term $\xi_1 \xi_2$ between the latent variables. Results show a positive relationship between economic performance and international asset dispersion, but country environment diversity is negatively associated

with performance, with a positive interaction between the two independent latent variables.

FUTURE TRENDS

The main future trend for the analysis of internationalization is the exploitation and development of more complex statistical techniques. Bayesian methods represent a natural evolution in this field. Hansen, Perry and Reese (2004) propose a Bayesian Hierarchical methodology to examine the relationship between administrative decisions and economic performance over time, which have the advantage of accounting for individual firm differences. The performance parameter is expressed as a function of the firm, the industry in which the firm operates and the set of administrative decisions (actions) made by the firm. More formally, let Y_{it} indicate the economic performance for company i ($i=1,...,175$) in year t ($t=1,...,4$), the

$$Y_{it} \approx N(\mu_{it} \sigma_t^2) \qquad (10)$$

meaning that Y_{it} is normally distributed, with mean μ_{it} and variance σ_t^2.

If X_{itk} ($k=1,...,10$) denotes the actions, the model is:

$$\mu_{it} = \eta_{it} + \sum_{k=1}^{10} \beta_{tk} X_{itk} \qquad (11)$$

where each firm is given its own effect.

A *prior* distribution must be specified for the entire set of parameters and the joint *posterior* distribution is then estimated with Markov Chain Monte Carlo (MCMC).

In the applied Bayesian model, the firm effect on economic performance can be isolated and indicates if the focal firm possesses some competitive advantage that will allow that firm to achieve economic performance greater than what would be predicted by the actions taken by that firm.

The advantages of Bayesian models are also underlined by Hahn and Doh (2006), showing that these methodologies are highly relevant not only for strategic problems, but also to its extensions in the areas of dynamic capabilities and co-evolution of industries and firms. The advantages of Bayesian methods include the full estimation of the distribution of individual effect terms, a straightforward estimation of predictive results even in complex models and exact distributional results under skew and small samples.

Another expected future trend could be represented by Bayesian Networks, which have the advantages of the Bayesian Models and belong also to the class of causal models (for further details about Bayesian Networks, see Bernardo and Smith, 1994).

CONCLUSION

In the previous sections we have seen how we can apply statistical techniques such as regression and causal models to understand the determinants of firm performance.

Regression methods are easier to implement, since they categorize variables simply into dependent and independent and they do not allow causal links among them. On the other hand, causal methods are more complex, since the presence of causal links allows us to model more flexible relationships among exogenous and endogenous variables.

The common objective of these models is to show how the dependent variable is related to the covariates, but while regression aims to reduce the number of independent variables influencing the dependent variable, causal models aim to represent correlations in the most accurate way. Given the high level of complexity of the processes of firms' international expansion, these statistical techniques seem well-suited to understanding and explaining the complex web of factors that impact on a firms' international performance. Therefore, with causal models we do not only implement an explorative analysis, which characterize regression methods, but we can also show causal ties, distinguishing them from the spurious ones.

International Business represents a new field for Data Mining researchers, which so far focused their attention on classical methodologies. But there is space for innovative applications, as for example Bayesian models, to improve the results in this area.

ACKNOWLEDGMENT

I would like to thank MUSING MUlti-industry, Semantic-based next generation business INtelliGence - Project IST-27097 and the DMPMI project (Modelli

di data mining e knowledge management per le piccole e medie imprese), FIRB 2003 D.D. 2186-Ric 12/12/2003.

REFERENCES

Basile, R., Castellani, D. & Zanfei A. (2003). Location choices of multinational firms in Europe: the role of national boundaries and EU policy, *Faculty of Economics, University of Urbino*, Working Paper n.183.

Beamish, P. W. & Ruihua, J. (2002). Investing profitably in China: Is it getter harder?, *Long Range Planning*, vol. 35(2), pp. 135-151.

Bernardo, J. M. & Smith, A. F. M. (1994). *Bayesian Theory*. Chichester. Wiley.

Bollen, K. A. (1989). *Structural Equations with Latent Variables*, New York, Wiley.

Chang, S-J.& Park S. (2005). Types of firms generating network externalities and MNCs co-location decisions, *Strategic Management Journal*, vol. 26, pp. 595-615.

Contractor, F. J., Kundu, S. K. & Hsu C. (2003). A three-stage theory of international expansion: the link between multinationality and performance in the service sector. *Journal of International Business Studies*, vol. 34 (1), pp. 5-18.

Cramer, J. S., (2003). *Logit models from economics and other fields*, Cambridge, Cambridge University Press.

Deshpande, R. & Zaltman, G. (1982). Factors Affecting the Use of Market Research Information: A Path Analysis, *Journal of Marketing Research*, Vol. XIX, February

Dillon, W. R. & Goldstein, M. (1984). *Multivariate Analysis: Methods and Applications,* Wiley.

Ford, S. & Strange R. (1999). Where do Japanese manufacturing firms invest within Europe, and why?, *Transnational Corporations*, vol. 8(1), pp. 117-142.

Giudici, P. (2003). *Applied Data Mining*, London, Wiley.

Goerzen, A. & Beamish, P. W. (2003). Geographic Scope and Multinational Enterprise Performance, *Strategic management Journal*, vol. 24, pp. 1289-1306.

Goldberger, A. S. (1973). *Structural equation models in the social sciences*, New York, 1973.

Greene, W. H. (1997). *Econometric Analysis* (third edition), New Jersey, Prentice Hall.

Hahn, E. D. & Doh, J. P. (2006). Using Bayesian Methods in Strategy Research: An Extension of Hansen et al. *Strategic Management Journal*, vol. 27, pp. 783-798.

Hansen, M. H., Perry, L. T. & Reese, C. S. (2004). A Bayesian Operationalization of the Resource-Based View, *Strategic Management Journal*, vol. 25, pp. 1279-1295.

Hensher, D. A., Rose, J. M. & Greene, W. H. (2005). *Applied choice analysis: a primer*, Cambridge, Cambridge University Press.

Hitt, M. A., Hoskisson, R. E. & Hicheon, K. (1997). International diversification: effects on innovation and firm performance in product-diversified firms, *The Academy of Management Journal*, vol. 40, n. 4, pp. 767-798.

Kutner, M. H., Nachtsheim, C. J. & Neter, J. (2004). *Applied Linear Regression Models*, McGraw-Hill, New York.

Loehlin, J. C. (1998). *Latent Variable Models: an introduction to factor, Path and Structural Analysis*, Mahwah (N. J.), London, Erlbaum.

Louviere, J., Hensher, D. & Swait, J. (2000). *Stated choice method. Analysis and Application.* Cambridge University Press, Cambridge.

Mardia, K. V., Kent J. T. & Bibby, J. M. (1979). *Multivariate Analysis.* Academic Press: London.

McFadden, D. (1974). Conditional logit analysis of qualitative choice behavior, in Zarembka P. (ed) *Frontiers in econometrics*, pp.105-142, New York, Academic Press.

Mueller, R. O. (1996). Linear Regression And Classical Path Analysis, in: *Basic Principles Of Structural Equation Modelling*. Springer Verlag.

Train, K. E. (2003). *Discrete choice methods with simulation*, Cambridge University Press,Cambridge.

KEY TERMS

Exogenous and Endogenous Variables: Exogenous variables are always independent variables without a causal antecedent; endogenous variables can be mediating variables (causes of other variables) or dependent variables.

Indicators: Observed variables, which determine latent variables in SEM.

Logistic Regression: Statistical method, belonging to the family of generalized linear models, whose principal aim is to model the link between a discrete dependent variable and some independent variables.

Multiple Regression: Statistical model that estimates the conditional expected value of one variable y given the values of some other variables x_i. The variable of interest, y, is conventionally called the *dependent variable* and the other variables are called the *independent variables*.

Path Analysis: Extension of the regression model belonging to the family of causal models. The most important characteristic of this technique is the presence of causal links among the variables.

Spurious Correlation: Association between two endogenous variables caused by an exogenous variable.

Structural Equation Modeling: Causal model that combines the characteristics of factor and path analysis, since it allows the presence of causal ties and latent variables.

Data Mining for Lifetime Value Estimation

Silvia Figini
University of Pavia, Italy

INTRODUCTION

Customer lifetime value (LTV, see e.g. Bauer et al. 2005 and Rosset et al. 2003), which measures the profit generating potential, or value, of a customer, is increasingly being considered a touchstone for administering the CRM (Customer relationship management) process. This in order to provide attractive benefits and retain high-value customers, while maximizing profits from a business standpoint. Robust and accurate techniques for modelling LTV are essential in order to facilitate CRM via LTV. A customer LTV model needs to be explained and understood to a large degree before it can be adopted to facilitate CRM. LTV is usually considered to be composed of two independent components: tenure and value. Though modelling the value (or equivalently, profit) component of LTV, (which takes into account revenue, fixed and variable costs), is a challenge in itself, our experience has revealed that finance departments, to a large degree, well manage this aspect. Therefore, in this paper, our focus will mainly be on modelling tenure rather than value.

BACKGROUND

A variety of statistical techniques arising from medical survival analysis can be applied to tenure modelling (i.e. semi-parametric predictive models, proportional hazard models, see e.g. Cox 1972). We look at tenure prediction using classical survival analysis and compare it with data mining techniques that use decision tree and logistic regression. In our business problem the survival analysis approach performs better with respect to a classical data mining predictive model for churn reduction (e.g. based on regression or tree models). In fact, the key challenge of LTV prediction is the production of segment-specific estimated tenures, for each customer with a given service supplier, based on the usage, revenue, and sales profiles contained in company databases. The tenure prediction models we have developed generate, for a given customer i, a hazard curve or a hazard function, that indicates the probability $h_i(t)$ of cancellation at a given time t in the future. A hazard curve can be converted to a survival curve or to a survival function which plots the probability $S_i(t)$ of "survival" (non-cancellation) at any time t, given that customer i was "alive" (active) at time (t-1), i.e., $S_i(t)=S_i(t-1) \times [1-h_i(t)]$ with $S_i(1)=1$. Once a survival curve for a customer is available, LTV for that specific customer i can be computed as:

$$\text{LTV} = \sum_{t=1}^{T} S_i(t) \times v_i(t), \qquad (1)$$

where $v_i(t)$ is the expected value of customer i at time t and T is the maximum time period under consideration. The approach to LTV (see e.g. Berger et. Al. 1998) computation provides customer specific estimates (as opposed to average estimates) of the total expected future (as opposed to past) profit based on customer behaviour and usage patterns. In the realm of CRM, modelling customer LTV has a wide range of applications including:

- Evaluating the returns of the investments in special offers and services.
- Targeting and managing unprofitable customers.
- Designing marketing campaigns and promotional efforts
- Sizing and planning for future market opportunities

Some of these applications would use a single LTV score computed for every customer. Other applications require a separation of the tenure and value component for effective implementation, while even others would use either the tenure or value and ignore the other component. In almost all cases, business analysts who use LTV are most comfortable when the predicted LTV score and/or hazard can be explained in intuitive terms.

Our case study concerns a media service company. The main objective of such a company is to maintain

its customers, in an increasingly competitive market; and to evaluate the lifetime value of such customers, to carefully design appropriate marketing actions. Currently the company uses a data mining model that gives, for each customer, a probability of churn (score).

The churn model used in the company to predict churn is currently a classification tree (see e.g. Giudici 2003). Tree models can be defined as a recursive procedure, through which a set of n statistical units is progressively divided in groups, according to a divisive rule which aims to maximize a homogeneity or purity measure of the response variable in each of the obtained groups. Tree models may show problems in time-dependent applications, such as churn applications.

MAIN FOCUS

The use of new methods is necessary to obtain a predictive tool which is able to consider the fact that churn data is ordered in calendar time. To summarize, we can sum up at least four main weaknesses of traditional models in our set-up, which are all related to time-dependence:

- excessive influence of the contract deadline date
- redundancies of information
- presence of fragmentary information, depending on the measurement time
- excessive weight of the different temporal perspectives

The previous points explain why we decided to look for a novel and different methodology to predict churn.

Future Survival Analysis Models to Estimate Churn

We now turn our attention towards the application of methodologies aimed at modelling survival risks (see e.g. Klein and Moeschberger 1997). In our case study the risk concerns the value that derives from the loss of a customer. The objective is to determine which combination of covariates affect the risk function, studying specifically the characteristics and the relation with the probability of survival for every customer.

Survival analysis (see e.g. Singer and Willet 2003) is concerned with studying the time between entry to a study and a subsequent event (churn). All of the standard approaches to survival analysis are probabilistic or stochastic. That is, the times at which events occur are assumed to be realizations of some random processes. It follows that T, the event time for some particular individual, is a random variable having a probability distribution. A useful, model-free approach for all random variables is nonparametric (see e.g. Hougaard 1995), that is, using the cumulative distribution function. The cumulative distribution function of a variable T, denoted by F(t), is a function that tell us the probability that the variable will be less than or equal to any value t that we choose. Thus, $F(t) = P\{T \leq t\}$. If we know the value of F for every value of t, then we know all there is to know about the distribution of T. In survival analysis it is more common to work with a closely related function called the survivor function defined as $S(t) = P\{T > t\} = 1 - F(t)$. If the event of interest is a death (or, equivalently, a churn) the survivor function gives the probability of surviving beyond t. Because S is a probability we know that it is bounded by 0 and 1 and because T cannot be negative, we know that S(0) =1. Finally, as t gets larger, S never increases. Often the objective is to compare survivor functions for different subgroups in a sample (clusters, regions…). If the survivor function for one group is always higher than the survivor function for another group, then the first group clearly lives longer than the second group.

When variables are continuous, another common way of describing their probability distributions is the probability density function. This function is defined as:

$$f(t) = \frac{dF(t)}{dt} = -\frac{dS(t)}{dt}, \qquad (2)$$

that is, the probability density function is just the derivative or slope of the cumulative distribution function. For continuous survival data, the hazard function is actually more popular than the probability density function as a way of describing distributions. The hazard function (see e.g. Allison 1995) is defined as:

$$h(t) = \lim \varepsilon t \to 0 \frac{\Pr\{t \leq T < t + \varepsilon t \mid T \geq t\}}{\varepsilon t}, \qquad (3)$$

The aim of the definition is to quantify the instantaneous risk that an event will occur at time t. Since time

is continuous, the probability that an event will occur at exactly time t is necessarily 0. But we can talk about the probability that an event occurs in the small interval between t and t+εt and we also want to make this probability conditional on the individual surviving to time t. For this formulation the hazard function is sometimes described as a conditional density and, when events are repeatable, the hazard function is often referred to as the intensity function. The survival function, the probability density function and the hazard function are equivalent ways of describing a continuous probability distribution. Another formula expresses the hazard in terms of the probability density function:

$$h(t) = \frac{f(t)}{S(t)}, \qquad (4)$$

and together equations (4) and (2) imply that

$$h(t) = -\frac{d}{dt}\log S(t) \quad (5),$$

Integrating both sides of equation (5) gives an expression for the survival function in terms of the hazard function:

$$S(t) = \exp\left(-\int_0^t h(u)du\right), \qquad (6)$$

With regard to numerical magnitude, the hazard is a dimensional quantity that has the form: number of events per interval of time. The database available for our analysis contain information that can affect the distribution of the event time, as the demographic variables, variables about the contract, the payment, the contacts and geo-marketing.

We remind the readers that the target variable has a temporal nature and, for this reason, it is preferable to build predictive models through survival analysis.

The actual advantages of using a survival analysis approach with respect to a traditional one can be reassumed in following:

- to correctly align the customers regarding their cycle of life;
- to analyze the real behaviour of the customers churn, without having to distinguish between EXIT and SUSPENSION payers.

In order to build a survival analysis model, we have constructed two variables: one variable of status (distinguish between active and non active customers) and one of duration (indicator of customer seniority). The first step in the analysis of survival data (for the descriptive study) consists in a plot of the survival function and the risk. The survival function is estimated through the methodology of Kaplan Meier (see e.g. Kaplan Meier 1958). Suppose there are K distinct event times, $t_1 < t_2 < \ldots < t_k$. At each time t_j there are n_j individuals who are said to be at risk of an event. At risk means they have not experienced an event not have they been censored prior to time t_j. If any cases are censored at exactly t_j, there are also considered to be at risk at t_j. Let d_j be the number of individuals who die at time t_j. The KM estimator is defined as:

$$\hat{S}(t) = \prod_{j:t_j \le t} [1 - \frac{d_j}{n_j}] \ (5.7) \text{ for } t_1 \le t \le t_k, \qquad (7)$$

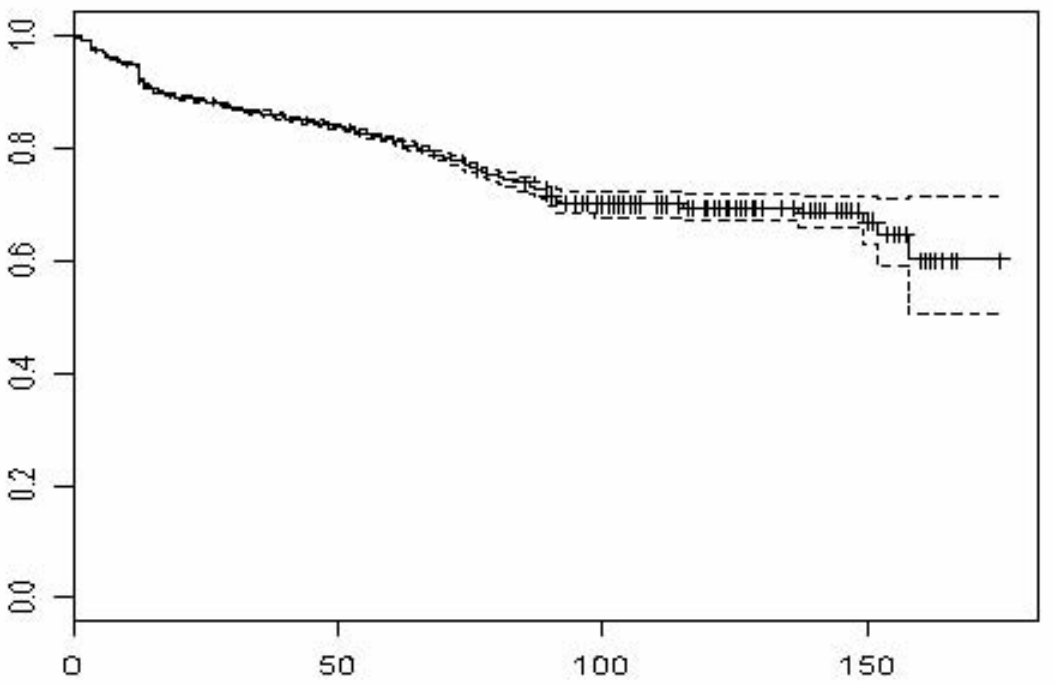
This formula says that, for a given time t, take all the event times that are less than or equal to t. For each of those event times, compute the quantity in brackets, which can be interpreted as the conditional probability of surviving to time t_{j+1}, given that one has survived to time t_j. Then multiply all of these survival probabilities together.

We now consider our application to the company data. Figure 1 shows the survival function for the whole customer database. On the x-axis there is the survival time (in months) and in the vertical axis we can see the probability of survival.

From Figure 1 note that the survival function has varying slopes, corresponding to different periods. When the curve decreases rapidly we have time periods with high churn rates; when the curve decreases softly we have periods of "loyalty". We remark that the final jump is due to a distortion caused by a few data, in the tail of the lifecycle distribution.

A very useful information, in business terms, is the calculation of the life expectancy of the customers. This can be obtained as a sum over all observed event times: $\hat{S}(t_{(j)})(t_{(j)} - t_{(j-1)})$, where $\hat{S}(t_{(j)})$ is the estimate of the survival function at the j-th event time, obtained using Kaplan-Meier method, and *t* is a duration indicator. ($t_{(0)}$ is by assumption equal to 0). We remark that life expectancy tends to be underestimated if most observed event types are censored (i.e., no more observable).

Figure 1. Descriptive survival function



Survival Predictive Model

We have chosen to implement Cox's model (see e.g. Cox 1972). Cox proposed a model that is a proportional hazards model and also he proposed a new estimation method that was later named partial likelihood or more accurately, maximum partial likelihood. We will start with the basic model that does not include time-dependent covariate or non proportional hazards. The model is usually written as:

$$h(t_{ij}) = h_0(t_j)\exp[\beta_1 X_{1ij} + b_2 X_{2ij} + ... + \beta_P X_{Pij}], \quad (8)$$

Equation 8 says that the hazard for individual i at time t is the product of two factors: a baseline hazard function that is left unspecified, and a linear combination of a set of p fixed covariates, which is then exponentiated. The baseline function can be regarded as the hazard function for an individual whose covariates all have values 0. The model is called proportional hazard model because the hazard for any individual is a fixed proportion of the hazard for any other individual. To see this, take the ratio of the hazards for two individuals i and j:

$$\frac{h_i(t)}{h_j(t)} = \exp\{\ \beta_1 (x_{i1} - x_{j1}) + \ldots + \beta_p (x_{ip} - x_{jp})\ \}, \quad (9)$$

What is important about this equation is that the baseline cancels out of the numerator and denominator. As a result, the ratio of the hazards is constant over time.

In Cox model building the objective is to identify the variables that are more associated with the churn event. This implies that a model selection exercise, aimed at choosing the statistical model that best fits the data, is to be carried out. The statistical literature presents many references for model selection (see e.g. Giudici 2003, Anderson 1991).

Most models are based on the comparison of model scores. The main score functions to evaluate models are related to the Kullback-Leibler principle (see e.g. Spiegelhalter et al. 2006).

This occurs for criteria that penalize for model complexity, such as AIC (Akaike Information Criterion, see e.g. Akaike et al., 1974) and BIC (Bayesian Information Criterion). In formal terms the AIC criterion is defined by the following equation:

$$AIC = -2\log L(\hat{\vartheta}; x_1,...,x_n) + 2q, \quad (10)$$

where $\log L(\hat{\vartheta}; x_1,...,x_n)$ is the logarithm of the likelihood function, calculated in the maximum likelihood parameter estimate and q is the number of parameters of the model. Notice that the AIC score essentially penalises the log-likelihood score with a term that increases linearly with model complexity.

The AIC criterion is based on the implicit assumption that q remains constant when the size of the sample increases. However this assumption is not always valid and therefore the AIC criterion does not lead to a consistent estimate of the dimension of the unknown model.

An alternative, and consistent, scoring function is the BIC criterion, also called SBC. It was formulated by Schwarz (1978) and is defined by the following expression:

$$BIC = -2\log L(\hat{\vartheta}; x_1,...,x_n) + q\log(n), \quad (11)$$

As is easily seen the BIC differs from the AIC (see e.g. Burnham et. al., 2004) only in the second part which now also depends on the sample size n. To conclude, the scoring function criteria for selecting models which we have examined are easy to calculate and lead to a total ordering of the models.

In our case we have compared all models and chose the one with the lowest value of AIC and BIC. The obtained model presents a good fit, as shown in Table 1.

From Table 1 note that both AIC and BIC present lower values with the inclusion of covariates: this means that adding covariates lead to a better fit (see e.g. Spitzner, 2006).

The result of the procedure is a set of about twenty explanatory variables. Such variables can be grouped in three main categories, according to the sign of their association with the churn rate, represented by the hazard ratio:

Table 1. Goodness of fit statistics

Model Fit Statistics		
Criterion	Without Covariates	With Covariates
-2 LOG L	1124678.6	943660.62
AIC	1124678.6	943788.62
BIC	1124678.6	944347.95

- variables that show a positive association (e.g. related to the wealth of the geographic region, the quality of the call center service, the sales channel)
- variables that show a negative association (e.g. number of technical problems, cost of service bought, status of payment method)
- variables that have no association (e.g. equipment rental cost, age of customer, number of family components).

More precisely, to calculate the previous associations we have considered the values of the hazard ratio under different covariate values. For example, for the variables indicating number of technical problems we compare the hazard function for those that have called at least once with those that have not made such calls. As the resulting ratio turns out to be equal to 0.849, the risk of becoming churner is lower for the "callers" than for the "non callers".

FUTURE TRENDS

Our results show that survival analysis modelling is a much powerful tool for lifetime value analysis and, consequently, for the actual planning of a range of marketing actions that impact on both perspective and actual customers. We believe that survival analysis is a very promising tool in the area, and that further research is indeed needed, both in applied and methodological terms. From an applied viewpoint, directions to be further investigated concern the application of

the methodology to a wider range of companies (we have studies in progress in the banking sector). From a methodological viewpoint further research is needed on the robustification of Cox model which, being semi-parametric and highly structured, may lead to variable results.

CONCLUSION

In the paper we have presented a comparison between classical and novel data mining techniques to predict rates of churn of customers. Our conclusions show that the novel approach we have proposed, based on survival analysis modelling, leads to more robust conclusions. In particular, although the lift of the best models are substantially similar, survival analysis modelling gives more valuable information, such as a whole predicted survival function, rather than a single predicted survival probability.

REFERENCE

Akaike, Hirotugu (December 1974). "A new look at the statistical model identification". IEEE Transactions on Automatic Control 19(6): 716–723.

Allison, P. D. (1995). *Survival Analysis Using the SAS Sysrem*, Cary, NC SAS Institute

Anderson, K.M., (1991). *A non proportional hazards Weibull accelerated failure time regression model.* Biometrics 47, 281-288.

Bauer, Hans H. and Maik Hammerschmidt (2005), "Customer-Based Corporate Valuation – Integrating the Concepts of Customer Equity and Shareholder Value," Management Decision, 43 (3), 331-348.

Berger, Paul D. and Nada I. Nasr (1998), "Customer lifetime value: Marketing models and applications," Journal of Interactive Marketing, 12 (1), 17 – 30.

Burnham, K.P., and D.R. Anderson. 2004. Multimodel Inference: understanding AIC and BIC in Model Selection, Amsterdam Workshop on Model Selection.

Cox, D. R. (1972). *Regression Models and Life Tables*, Journal of the Royal Statistical Society, B34, 187-220.

Giudici, P. (2003). *Applied Data Mining*, Wiley.

Hougaard, P. (1995). *Frailty models for survival data.* Lifetime Data Analysis. 1: 255-273.

Kaplan, E. L. and Meier, R. (1958). *Nonparametric Estimation from Incomplete Observations*, Journal of the American Statistical Association, 53, 457-481.

Klein, J.P. and Moeschberger, M.L. (1997). *Survival analysis: Techniques for censored and truncated data.* New York: Springer.

Rosset, Saharon, Nuemann, Einat, Eick, Uri and Vatnik, Nurit (2003) Customer lifetime value models for decision support *Data Mining and Knowledge Discovery*, 7, 321-339.

Singer and Willet, (2003). *Applied Longitudinal Data Analysis*, Oxford University Press.

Spiegelhalter, D. and Marshall, E. (2006) Strategies for inference robustness in focused modelling *Journal of Applied Statistics*, 33, 217-232.

Spitzner,D. J. (2006) Use of goodness-of-fit procedures in high dimensional testing *Journal of Statistical Computation and Simulation*, 76, 447-457.

KEY TERMS

Customer Lifetime Value: (also variously referred to as lifetime customer value or just lifetime value, and abbreviated CLV, LCV, or LTV) is a marketing metric that projects the value of a customer over the entire history of that customer's relationship with a company.

Customer Relationship Management (CRM): includes the methodologies, technology and capabilities that help an enterprise manage customer relationships. The general purpose of CRM is to enable organizations to better manage their customers through the introduction of reliable systems, processes and procedures.

Geomarketing: A set of scientific methods, which is capable of collecting and working out data, allowing the evaluation of the customer's impact before opening a new sales point, the exact definition of the interested territory and the identification of ideal users for the distribution of advertising material based on specified

times and distances, avoiding superimpositions with neighboring user basins.

Hazard Risk: The hazard rate (the term was first used by Barlow, 1963) is defined as the probability per time unit that a case that has survived to the beginning of the respective interval will fail in that interval.

Robustification: Robust statistics have recently emerged as a family of theories and techniques for estimating the parameters of a parametric model while dealing with deviations from idealized assumptions. There are a set of methods to improve robustification in statistics, as: M-estimator, Least Median of Squares Estimator, W-estimators.

Data Mining for Model Identification

Diego Liberati
Italian National Research Council, Italy

INTRODUCTION

In many fields of research, as well as in everyday life, it often turns out that one has to face a huge amount of data, without an immediate grasp of an underlying simple structure, often existing. A typical example is the growing field of bio-informatics, where new technologies, like the so-called Micro-arrays, provide thousands of gene expressions data on a single cell in a simple and fast integrated way. On the other hand, the everyday consumer is involved in a process not so different from a logical point of view, when the data associated to his fidelity badge contribute to the large data base of many customers, whose underlying consuming trends are of interest to the distribution market.

After collecting so many variables (say gene expressions, or goods) for so many records (say patients, or customers), possibly with the help of wrapping or warehousing approaches, in order to mediate among different repositories, the problem arise of reconstructing a synthetic mathematical model capturing the most important relations between variables. To this purpose, two critical problems must be solved:

1 To select the most salient variables, in order to reduce the dimensionality of the problem, thus simplifying the understanding of the solution
2 To extract underlying rules implying conjunctions and/or disjunctions between such variables, in order to have a first idea of their even non linear relations, as a first step to design a representative model, whose variables will be the selected ones

When the candidate variables are selected, a mathematical model of the dynamics of the underlying generating framework is still to be produced. A first hypothesis of linearity may be investigated, usually being only a very rough approximation when the values of the variables are not close to the functioning point around which the linear approximation is computed.

On the other hand, to build a non linear model is far from being easy: the structure of the non linearity needs to be a priori known, which is not usually the case. A typical approach consists in exploiting a priori knowledge to define a tentative structure, and then to refine and modify it on the training subset of data, finally retaining the structure that best fits a cross-validation on the testing subset of data. The problem is even more complex when the collected data exhibit hybrid dynamics, i.e. their evolution in time is a sequence of smooth behaviors and abrupt changes.

BACKGROUND

Such tasks may be sequentially accomplished with various degree of success in a variety of ways. Principal components (O'Connel 1974) orders the variables from the most salient to the least one, but only under a linear framework (Liberati et al., 1992a).

Partial least squares (Dijkstra 1983) allow to extend to non linear models, provided that one has some a priori information on the structure of the involved non linearity: in fact, the regression equation needs to be written before identifying its parameters. Clustering may operate in an unsupervised way, without the a priori correct classification of a training set (Booley 1998).

Neural networks are known to learn the embedded rules, with the indirect possibility (Taha and Ghosh 1999) to make rules explicit or to underline the salient variables. Decision trees (Quinlan 1994) are a popular framework providing a satisfactory answer to both questions.

Systems identification (Söderström and Stoica, 1989) is widely and robustly addressed even in calculus tools like Matlab from Matworks.

MAIN FOCUS

More recently, a different approach has been suggested, named Hamming Clustering (Muselli & Liberati 2000). It is related to the classical theory exploited in minimizing the size of electronic circuits, with the additional

care to obtain a final function able to generalize from the training data set to the most likely framework describing the actual properties of the data. Such approach enjoys the following remarkable two properties:

a) It is fast, exploting, after proper binary coding, just logical operations instead of floating point multiplications

b) It directly provides a logical understandable expression (Muselli & Liberati 2002), being the final synthesized function directly expressed as the OR of ANDs of the salient variables, possibly negated.

An alternative approach is to infer the model directly from the data via an identification algorithm capable to reconstruct a very general class of piece-wise affine models (Ferrari-Trecate et al., 2003). This method can be also exploited for the data-driven modeling of hybrid dynamical systems where logic phenomena interact with the evolution of continuous-valued variables.. Such approach will be concisely described later in the following, after a little more detailed drawing of the rules-oriented mining procedure. The last section will briefly discuss some applications.

HAMMING CLUSTERING: BINARY RULE GENERATION AND VARIABLE SELECTION WHILE MINING DATA

The approach followed by Hamming Clustering (HC) in mining the available data to select the salient variables and to build the desired set of rules consists of the three steps in Table 1.

- **Step 1:** A critical issue is the partition of a possibly continuous range in intervals, whose number and limits may affect the final result. The thermometer code may then be used to preserve ordering and distance (in case of nominal input variables, for which a natural ordering cannot be defined, the only-one code may instead be adopted). The metric used is the Hamming distance, computed as the number of different bits between binary strings: the training process does not require floating point computation, but only basic logic operations, for the sake of speed and insensitivity to precision.
- **Step 2:** Classical techniques of logical synthesis, such as ESPRESSO (Brayton at al 1984) or MINI (Hong 1997), are specifically designed to obtain the simplest AND-OR expression able to satisfy all the available input-output pairs, without an explicit attitude to generalize. To generalize and infer the underlying rules, at every iteration HC groups together, in a competitive way, binary strings having the same output and close to each other. A final pruning phase does simplify the resulting expression, further improving its generalization ability. Moreover, the minimization of the involved variables do intrinsically exclude the redundant ones, thus enhancing the very salient variables for the investigated problem. The quadratic computational cost allows to manage quite large datasets.
- **Step 3:** Each logical product directly provides an intelligible rule synthesizing a relevant aspect of the searched underlying system that is believed to generate the available samples.

Table 1. The tree steps executed by Hamming clustering to build the set of rules embedded in the mined data

1.	The input variables are converted into binary strings via a coding designed to preserve distance and, if relevant, ordering.
2.	The 'OR of ANDs' expression of a logical function is derived from the training examples coded in the binary form of step 1
3.	In the OR final expression, each logical AND provides intelligible conjunctions or disjunctions of the involved variables, ruling the analyzed problem

IDENTIFICATION OF PIECEWISE AFFINE SYSTEMS THROUGH A CLUSTERING TECHNIQUE

Once the salient variables have been selected, it may be of interest to capture a model of their dynamical interaction. Piecewise affine (PWA) identification exploits *K-means* clustering that associates data points in multivariable space in such a way to jointly determine a sequence of linear sub-models and their respective regions of operation, without even imposing continuity at each change in the derivative. In order to obtain such result, the five steps reported in Table 2 are executed:

- **Step 1:** The model is locally linear: small sets of data points close to each other likely belong to the same sub-model. For each data point, a local set is built, collecting the selected point together with a given number of its neighbours (whose cardinality is one of the parameters of the algorithm). Each local set will be *pure* if made of points really belonging to the same single linear subsystem, otherwise *mixed.*
- **Step 2:** For each local data set, a linear model is identified through least squares procedure. *Pure* sets, belonging to the same sub-model, give similar parameter sets, while *mixed* sets yield isolated vectors of coefficients, looking as outliers in the parameter space. If the signal to noise ratio is good enough, and there are not too many mixed sets (i.e. the number of data points are enough more than the number of sub-models to be identified, and the sampling is fair in every region), the vectors will cluster in the parameter space around the values pertaining to each sub-model, apart from a few outliers.
- **Step 3:** A modified *K-means* (Ferrari-Trecate et al., 2003), whose convergence is guaranteed in a finite number of steps, takes into account the confidence on pure and mixed local sets, in order to cluster the parameter vectors.
- **Step 4:** Data points are then classified, being each local data set one-to-one related to its generating data point, which is thus classified according to the cluster to which its parameter vector belongs.
- **Step 5:** Both the linear sub-models and their regions are estimated from the data in each subset. The coefficients are estimated via weighted least squares, taking into account the confidence measures. The shape of the polyhedral region characterizing the domain of each model may be obtained either via Linear Support Vector Machines (Vapnik, 1998), easily solved via linear/quadratic programming, or via Multi-category Robust linear Programming (Bennet & Mangasarian, 1992).

FUTURE TRENDS

The field of application of the proposed approaches is wide, being both tools general and powerful, especially if combined together. A few suggestions will be drawn, with reference to the field of life science because of its relevance to science and society. Systems Biology, the edge of Bio-informatics, starts from analyzing data from Micro-arrays, where thousands of gene expressions may be obtained from the same cell material, thus providing a huge amount of data whose handling with the usual approaches is not conceivable, fault to obtain not significant synthetic information. Thus, matrices of subjects, possibly grouped in homogeneous categories for supervised training, each one carrying his luggage of thousands of gene expressions, are the natural input

Table 2. The five steps for piecewise affine identification

1.	Local data sets are built neighbouring each sample
2.	Local linear models are identified through least squares
3.	Parameters vectors are clustered through a modified *K-means*
4.	Data points are classified accordingly to the clustered parameter vectors
5.	Sub-models are estimated via least squares; their regions are estimated via support vector machines or multi-category robust linear programming

to algorithms like HC. A desired output is to extract rules classifying, for instance, patients affected by different tumors from healthy subjects, on the basis of a few identified genes (Garatti et al.,., 2007), whose set is the candidate basis for the piecewise linear model describing their complex interaction in such a particular class of subjects. Also without such a deeper inside in the cell, the identification of the prognostic factors in oncology is already at hand with HC (Paoli et al., 2000), also providing a hint about their interaction, which is not explicit in the outcome of a neural network (Drago et al., 2002). Drug design does benefit form the a priori forecasting provided by HC about hydrophilic behavior of the not yet experimented pharmacological molecule, on the basis of the known properties of some possible radicals, in the track of Manga et al., (2003), within the frame of computational methods for the prediction of 'drug-likeness' (Clark & Pickett, 2000). Such in silico predictions, is relevant to Pharmaceutical Companies in order to save money in designing minimally invasive drugs whose kidney metabolism would less affect the liver the more hydrophilic they are.

Compartmental behaviour of drugs may then be analysed via piece-wise identification by collecting in vivo data samples and clustering them within the more active compartment at each stage, instead of the classical linear system identification that requires non linear algorithms (Maranzana et al., 1986). The same happens for metabolism, like glucose in diabetes or urea in renal failure: dialysis can be modelled as a compartmental process in a sufficiently accurate way (Liberati et al., 1993). A piece-wise linear approach is able to simplify the identification even on a single patient, when population data are not available to allow a simple linear de-convolution approach (Liberati & Turkheimer 1999) (whose result is anyway only an average knowledge of the overall process, without taking into special account the very subject under analysis). Compartmental model are such pervasive, like in ecology, wildlife and population studies, that potential applications in such direction are almost never ending.

Many physiological processes are switching form active to quiescent state, like Hormone pulses, whose identification (De Nicolao & Liberati 1993) is useful in growth and fertility diseases among others, as well as in doping assessment. In that respect, the most fascinating human organ is probably the brain, whose study may be undertaken today either in the sophisticated frame of functional Nuclear Magnetic Imaging or in the simple way of EEG recording. Multidimensional (3 space variables plus time) images or multivariate time series provide lot of raw data to mine in order to understand which kind of activation is produced in correspondence with an event or a decision (Baraldi et al., 2007). A brain computer interfacing device may then be built, able to reproduce one's intention to perform a move, not directly possible for the subject in some impaired physical conditions, and command a proper actuator. Both HC and PWA Identification do improve the already interesting approach based on Artificial Neural Networks (Babiloni et al., 2000). A simple drowsiness detector based on the EEG can be designed, as well as an anaesthesia/hypotension level detector more flexible than the one in (Cerutti et al., 85), without needing the time-varying more precise, but more costly, identification in (Liberati et al., 91a). A psychological stress indicator can be inferred, outperforming (Pagani et al., 1991). The multivariate analysis (Liberati et al., 1997), possibly taking into account also the input stimulation (Liberati et al., 1989, 1991b, 1992b,c), useful in approaching not easy neurological tasks like modelling Electro Encephalographic coherence in Alzheimer's disease patients (Locatelli et al., 1998) or non linear effects in muscle contraction (Orizio et al., 1996), can be outperformed by PWA identification even in time-varying contexts like epilepsy, aiming to build an implantable defribillator.

Industrial applications are of course not excluded from the field of possible interests. In (Ferrari-Trecate et al., 2003), for instance, the classification and identification of the dynamics of an industrial transformer are performed via the piece-wise approach, with a much simpler cost, not really paid in significant reduction of performances, with respect to the non linear modelling described in (Bittanti et al., 2001).

CONCLUSION

The proposed approaches are very powerful tools for quite a wide spectrum of applications, providing an answer to the quest of formally extracting knowledge from data and sketching a model of the underlying process.

Table 3. Summary of selected applications of the proposed approaches

System Biology and Bioinformatics	To identify the interaction of the main proteins involved in cell cycle
Drug design	To forecast the behavior of the final molecule from components
Compartmental modeling	To identify number, dimensions and exchange rates of communicating reservoirs
Hormone pulses detection	To detect true pulses among the stochastic oscillations of the baseline
Sleep detector	To forecast drowsiness, as well as to identify sleep stages and transitions
Stress detector	To identify the psychological state from multivariate analysis of biological signals
Prognostic factors	To identify the interaction between selected features, for instance in oncology
Pharmaco-kinetics and Metabolism	To identify diffusion and metabolic time constants from time series sampled in blood
Switching processes	To identify abrupt or possibly smoother commutations within the process duty cycle
Brain Computer Interfacing	To identify and propagate directly form brain waves a decision
Anesthesia detector	To monitor the level of anesthesia and provide close-loop control
Industrial applications	Wide spectrum of dynamic-logical hybrid problems, like tracking a transformer

ACKNOWLEDGMENT

Marco Muselli had a key role in conceiving and developing the clustering algorithms, while Giancarlo Ferrari-Trecate has been determinant in extending them to model identification

REFERENCES

Babiloni, F., Carducci, F., Cerutti, S., Liberati, D., Rossini, P., Urbano, A., & Babiloni, C. (2000). Comparison between human and ANN detection of laplacian-derived electroencephalographic activity related to unilateral voluntary movements. *Computers and Biomedical Research, 33*, 59-74.

Baraldi, P., Manginelli, A.A., Maieron, M., Liberati, D., & Porro, C.A. (2007). An ARX model-based approach to trial by trial identification of fMRI-BOLD responses**,** *NeuroImage, 37*, 189-201.

Bennet, K.P., & Mangasarian, O.L. (1994). Multicategory discrimination via linear programming. *Optimization Methods and Software, 4*, 27-39.

Bittanti, S., Cuzzola, F. A., Lorito, F., & Poncia, G. (2001). Compensation of nonlinearities in a current transformer for the reconstruction of theprimary current. *IEEE Transactions on Control System Technology*, *9*(4), 565-573.

Boley, D.L. (1998). Principal direction divisive partitioning. *Data Mining and Knowledge Discovery, 2*(4), 325-344.

Brayton, R.K., Hachtel, G.D., McMullen, C.T., and Sangiovanni-Vincentelli, A.L. (1984) *Logic Minimization Algorithms for VLSI Synthesis*.Hingham, MA: Kluwer Academic Publishers.

Cerutti, S., Liberati, D., & Mascellani, P. (1985). Parameters extraction in EEG Processing during riskful neurosurgical operations, *Signal Processing*, *9*, 25-35.

Clark, D.E., & Pickett, S.D. (2000). Computational methods for the prediction of 'drug-likeness', *Drug Discovery Today 5*(2), 49-58.

De Nicolao, G., and Liberati, D. (1993). Linear and nonlinear techniques for the deconvolution of hormone time-series, *IEEE Transactions in BioMedical Engineering, 40*(5), 440-455.

Dijkstra, T. (1983). Some comments on maximum likelihood and partial least squares methods, *Journal of Econometrics*, *22*, 67-90.

Drago, G.P., Setti, E., Licitra, L., & Liberati, D. (2002). Forecasting the performance status of head and neck cancer patient treatment by an interval arithmetic pruned perceptron*,IEEE Tranactions Biomedical Engineering, 49*(8) 782-787.

Ferrari Trecate, G., Muselli, M., Liberati, D., Morari, M. (2003). A clustering technique for the identification of piecewise affine systems, *Automatica* 39, 205-217

Garatti, S., Bittanti, S., Liberati, D., & Maffezzoli, P. (2007). An unsupervised clustering approach for leukemia classification based on DNA micro-arrays data, *Intelligent Data Analysis*, *11*(2),175-188.

Hong, S.J. (1997). R-MINI: An Iterative Approach for Generating Minimal Rules from Examples, *IEEE Tranactions on Knowledge and Data Engineering*, *9*, 709-717.

Liberati, D., Cerutti, S., DiPonzio, E., Ventimiglia, V., & Zaninelli, L. (1989). Methodological aspects for the implementation of ARX modelling in single sweep visual evoked potentials analysis, *Journal of Biomedical Engineering 11*, 285-292.

Liberati, D., Bertolini, L., & Colombo, D.C. (1991a). Parametric method for the detection of inter and intra-sweep variability in VEP's processing, *Medical & Biological Engineering & Computing*, *29*, 159-166.

Liberati, D., Bedarida, L., Brandazza, P., & Cerutti, S. (1991b). A model for the Cortico-Cortical neural interaction in multisensory evoked potentials, *IEEE Transactions on Biomedical Engineering, 38*(9), 879-890.

Liberati, D., Brandazza, P., Casagrande, L., Cerini, A., & Kaufman, B. (1992a). Detection of Transient Single-Sweep Somatosensory Evoked potential changes via principal component analysis of the autoregressive-with-exogenous-input parameters, In *Proceedings of XIV IEEE-EMBS,* Paris (pp. 2454-2455).

Liberati, D., Narici, L., Santoni, A., & Cerutti, S. (1992b). The dynamic behavior of the evoked magnetoencephalogram detected through parametric identification, *Journal of Biomedical Engineering*, *14*, 57-64.

Liberati, D., DiCorrado, S., & Mandelli, S. (1992c). Topographic mapping of single-sweep evoked potentials in the brain, *IEEE Transactions on Biomedical Engineering*, *39*(9), 943-951.

Liberati, D., Biasioli, S., Foroni, R., Rudello, F., & Turkheimer, F. (1993). A new compartmental model approach to dialysis, *Medical & Biological Engineering & Computing, 31*, 171-179.

Liberati, D., Cursi, M., Locatelli, T., Comi, G., & Cerutti, S. (1997). Total and partial coherence of spontaneous and evoked EEG by means of multi-variable autoregressive processing, *Medical & Biological Engineering & Computing*, *35*(2), 124-130.

Liberati, D. and Turkheimer, F. (1999). Linear spectral deconvolution of catabolic plasma concentration decay in dialysis, *Medical & Biological Engineering & Computing, 37*, 391-395.

Locatelli, T., Cursi, M., Liberati, D., Franceschi, M., and Comi, G. (1998). EEG coherence in Alzheimer's disease, *Electroencephalography and clinical neurophysiology, 106*(3), 229-237.

Manga, N., Duffy, J.C., Rowe, P.H., Cronin, M.T.D., (2003). A hierarchical quantitative structure-activity relationship model for urinary excretion of drugs in humans as predictive tool for biotrasformation, quantitative structure-activity relationship, *Combinatorial Science*, *22*, 263-273.

Maranzana, M., Ferazza, M., Foroni, R., and Liberati, D. (1986). A modeling approach to the study of beta-adrenergic blocking agents kinetics, *Measurement in Clinical Medicine* (pp. 221-227). Edinburgh: IMEKO Press.

Muselli, M., and Liberati, D. (2000). Training digital circuits with Hamming Clustering. *IEEE Transactions on Circuits and Systems – I: Fundamental Theory and Applications, 47*, 513-527.

Muselli, M., and Liberati, D. (2002). Binary rule generation via Hamming Clustering. *IEEE Transactions on Knowledge and Data Engineering, 14*, 1258-1268.

O'Connel, M.J. (1974), Search program for significant variables, *Computer Physics Communication*, *8*, 49.

Orizio, C., Liberati, D., Locatelli, C., DeGrandis, D., & Veicsteinas, A. (1996). Surface mechanomyogram reflects muscle fibres twitches summation, *Journal of Biomechanics*, *29*(4), 475-481.

Pagani, M., Mazzuero, G., Ferrari, A., Liberati, D., Cerutti, S., Vaitl, D., Tavazzi, L., & Malliani, A. (1991). Sympatho-vagal interaction during mental stress: a study employing spectral analysis of heart rate variability in healty controls and in patients with a prior myocardial infarction, *Circulation*, *83*(4), 43-51.

Quinlan, J.R., (1994). *C4.5: Programs for Machine Learning*. San Francisco: Morgan Kaufmann

Söderström, P.S. (1989): *System Identification*, London: Prentice Hall.

Taha, I., and Ghosh, J. (1999). Symbolic interpretation of artificial neural networks, *IEEE Transactions on Knowledge and Data Engineering*, (vol. 11, pp. 448-463).

Vapnik, V. (1998). *Statistical learning theory*. New York: Wiley.

KEY TERMS

Bio-Informatics: The processing of the huge amount of information pertaining biology.

Hamming Clustering: A fast binary rule generator and variable selector able to build understandable logical expressions by analyzing the Hamming distance between samples.

Hybrid Systems: Their evolution in time is composed by both smooth dynamics and sudden jumps.

Micro-Arrays: Chips where thousands of gene expressions may be obtained from the same biological cell material.

Model Identification: Definition of the structure and computation of its parameters best suited to mathematically describe the process underlying the data.

Rule Generation: The extraction from the data of the embedded synthetic logical description of their relationships.

Salient Variables: The real players among the many apparently involved in the true core of a complex business.

Systems Biology: The quest of a mathematical model of the feedback regulation of proteins and nucleic acids within the cell.

Data Mining for Obtaining Secure E-Mail Communications

Mª Dolores del Castillo
Instituto de Automática Industrial (CSIC), Spain

Ángel Iglesias
Instituto de Automática Industrial (CSIC), Spain

José Ignacio Serrano
Instituto de Automática Industrial (CSIC), Spain

INTRODUCTION

Email is now an indispensable communication tool and its use is continually growing. This growth brings with it an increase in the number of electronic threats that can be classified into five categories according to their inner behavior: virus, trojans, pharming, spam, and phishing. Viruses, trojans and pharming threats represent an attack to the user's computer while the focus of attack of spam and phishing threats is mainly the user, that is, these last two threats involve a kind of intellectual attack.

A virus is a small program that replicates itself and inserts copies into other executable code or documents using e-mails as a means of transport. Trojans can not replicate themselves and they are used to open a network port giving other users a means of controlling the infected computer. Other more dangerous trojans are called spy programs (spyware) which wait until users visit some websites and then capture all the keys typed and mouse movements and make screenshots to obtain information. Pharming is a technique used to redirect users to illegitimate websites. These three threats, in spite of being present in e-mails, can be solved by an anti virus program.

The next two threats need e-mail filters to be solved and this chapter focuses on them: spam and phishing. Spam consists on the massive sending of unsolicited commercial e-mail to a large number of recipients. Unlike legitimate commercial e-mail, spam is sent without the explicit permission of the recipients. Spammers obtain e-mail addresses by different ways such as guessing common names at known domains or searching addresses in web pages. A report from the Commission of European Communities ("Communication from", 2004) shows that more than 25 percent of all e-mail currently received is spam. More recent reliable data shows that spam represents 60-80 percent of e-mail volume. Spam is widely recognized as one of the most significant problems facing the Internet today.

Spam has evolved to a new and dangerous form known as 'phishing'. Phishing differs from spam in that it is generated by a criminal intent on stealing personal data for financial gain ("Spyware", 2007). Phishing is the term used to describe emails which trick recipients into revealing their personal or their company's confidential information such as social security and financial account numbers, account passwords and other identity or security information.

According to Anti-Phishing Working Group ("June Phishing", 2006) the number of phishing reports has increased from 20,109 in May 2006 to 28,571 in June 2006 and it is the most ever recorded. Phishing attacks increase despite of the efforts of e-mail filters. Although only 0.001 percent of e-mail sent is responded to, this percentage is enough to return on the investment and keep the phishing industry alive. Further research has estimated that the costs of these phishing attacks on consumers in 2003 ranged from $500 million to an amazing $2.4 billion.

The early phishing attempts consisted on a link to a website which looked like a legitimate website, but in fact was an illegitimate website. The website address usually was not a domain, but simply an IP address, and the e-mails were often very poorly written, with bad grammar and spelling, and little attention to detail. Needless to say that phishing attacks have evolved with more convincing content and became harder to recognize. While a non-professional appearance such as a spelling error, a dubious URL, or a non-secure website are sure signs of a fraudulent phishing website, the lack of these features can no longer be used as a sure sign of a legitimate site (Green, 2005).

It is hard to successfully obtain bibliographical information in the scientific and marketing literature about techniques that aim to avoid spam and electronic fraud. This could be due to the features of these security systems, which should not be revealed in public documents for security reasons. This lack of information prevents improvements of criminals' attacks because spammers/phishers just do not know the methods used to detect and eliminate their attacks. It is also necessary to emphasize that there is little available commercial technology that shows an actual and effective solution for users and businesses.

Spam and phishing filters process e-mail messages and then choose where these messages have to be delivered. These filters can deliver spurious messages to a defined folder in the user's mailbox or throw messages away.

BACKGROUND

Filtering can be classified into two categories, origin-based filtering or content-based filtering, depending on the part of the message chosen by the filter as the focus for deciding whether e-mail messages are valid or illegitimate (Cunningham, 2003). Origin-based filters analyse the source of the e-mail, i.e. the domain name and the address of the sender (Mertz, 2002) and check whether it belongs to white (Randazzese, 2004) or black (Pyzor, 2002) verification lists.

Content-based filters aim to classify the content of e-mails: text and links. Text classifiers automatically assign document to a set of predefined categories (legitimate and illegitimate). They are built by a general inductive process that learns, from a set of preclassified messages, the model of the categories of interest (Sebastiani, 2002). Textual content filters may be differentiated depending on the inductive method used for constructing the classifier in a variety of ways:

Rule-based filters. Rules obtained by an inductive rule learning method consist of a premise denoting the absence or presence of keywords in textual messages and a consequent that denotes the decision to classify the message under a category. Filters that use this kind of filtering assign scores to messages based on fulfilled rules. When a message's score reaches a defined threshold, it is flagged as illegitimate. There are several filters using rules (Sergeant, 2003).

Bayesian filters. First, the probabilities for each word conditioned to each category (legitimate and illegitimate) are computed by applying the Bayes theorem, and a vocabulary of words with their associated probabilities is created. The filter classifies a new text into a category by estimating the probability of the text for each possible category C_j, defined as $P(C_j \mid text) = P(C_j) \cdot \Pi_i P(w_i \mid C_j)$, where w_i represents each word contained in the text to be classified. Once these computations have been carried out, the Bayesian classifier assigns the text to the category that has the highest probability value. Filter shown in ("Understanding Phishing", 2006) uses this technique.

Memory-based filters. These classifiers do no build a declarative representation of the categories. E-mail messages are represented as feature or word vectors that are stored and matched with every new incoming message. These filters use e-mail comparison as their basis for analysis. Some examples of this kind of filter are included in (Daelemans, 2001). In (Cunningham, 2003), case-based reasoning techniques are used.

Other textual content filters. Some of the content-based approaches adopted do not fall under the previous categories. Of these, the most noteworthy are the ones based on support vector machines, neural networks, or genetic algorithms. SVMs are supervised learning methods that look, among all the n-dimensional hyperplanes that separate the positive from the negative messages training, the hyperplane that separates the positive from the negative by the widest margin (Sebastiani, 2002). Neural network classifiers are nets of input units representing the words of the message to be classified, output units representing categories, and weighted edges connecting units represent dependence relations that are learned by backpropagation (Vinther, 2002). Genetic algorithms represent textual messages as chromosomes of a population that evolves by copy, crossover and mutation operators to obtain the textual centroid or prototypical message of each category (Serrano, 2007).

Non textual content-based filters or link-based filters detect illegitimate schemes by examining the links embedded in e-mails. This filtering can be done by using white-link and black-link verification lists (GoDaddy, 2006) or by appearance analysis (NetCraft, 2004) looking for well known illegitimate features contained in the name of the links.

Origin-based filters as well as link-based filters do not perform any data mining task from e-mails samples.

MAIN FOCUS

Most current filters described in the previous section achieve an acceptable level of classification performance, detecting 80%-98% of illegitimate e-mails. The main difficulty of spam filters is to detect false positives, i.e., the messages that are misidentified as illegitimate. Some filters obtain false positives in nearly 2% of the tests run on them. Although these filters are used commercially, they show two key issues for which there is no solution at present: 1) Filters are tested on standard sets of examples that are specifically designed for evaluating filters and these sets become obsolete within a short period of time, and 2) In response to the acceptable performance of some filters, spammers study the techniques filters use, and then create masses of e-mails intended to be filtered out, so that the content-based filters with the ability to evolve over time will learn from them. Next, the spammers/phishers generate new messages that are completely different in terms of content and format.

The evolution ability of some filters presents a double side: they learn from past mistakes but learning can lead to upgrade their knowledge base (words, rules or stored mesages) in an undirected manner so that the classification phase of the filter for incoming e-mails becomes inefficient due to the exhaustive search of words, rules or examples that would match the e-mail.

In order to design effective and efficient filtering systems, that overcome the drawbacks of the current filters, it is necessary to adopt an integrating approach that can make use of all possible patterns found by analysing all types of data present in e-mails: senders, textual content, and non textual content as images, code, links as well as the website content addressed by such links. Each type of data must be tackled by the mining method or methods most suitable for dealing with this type of data. This multistrategy and integrating approach can be applied to filtering both spam and phishing e-mails.

In (Castillo, 2006) a multistrategy system to identify spam e-mails is described. The system is composed of a heuristic knowledge base and a Bayesian filter that classifies the three parts of e-mails in the same way and then integrates the classification result of each part. The vocabulary used by the Bayesian filter for classifying the textual content is not extracted after a training stage but it is learned incrementally starting from a previously heuristic vocabulary resulting from mining invalid words (for example: vi@gra, v!agra, v1agra). This heuristic vocabulary includes word patterns that are invalid because their morphology does not meet the morphological rules of all the languages belonging to the Indo-European family. Invalid words are primarily conceived to fool spam filters. A finite state automata is created from a set of examples of well-formed words collected randomly from a set of dictionaries of several languages. Words taken from e-mails labelled as spam are not recognized by the automata as valid words, and are thus identified as misleading words. Then, an unsupervised learning algorithm is used to build a set of invalid word patterns and their descriptions according to different criteria. So, this spam system performs two textual data mining tasks: on the words by a clustering method and on the full content by a Bayesian filter.

Phishing filters use verification list (for origin and links) which are combined with the anti spam technology developed until now. Phishing filters based on the source and/or on the links contained in the body of the e-mail are very bounded. The sole possible way of filter evolution implies that the white or black lists used have to be continuously updated. However, phishers can operate between one updating and the following one and the user can receive a fraudulent mail. So, in this type of attacks and due to its potential danger, the analysis of the filter has to be focused on minimizing the number of false negatives, i.e., the messages erroneously assigned to the legitimate class, produced by the filter. Anyway, building a phishing filter actually effective means that the filter should take advantage of a multistrategy data mining on all kind of data included in e-mails. Usually, phishing e-mails contain forms, executable code or links to websites. The analysis of these features can give arise to patterns or rules that the filtering system can look up in order to make a classification decision. These mined rules can complement a Bayesian filter devoted to classify the textual content.

A key issue missed in any phishing filter is the analysis of the textual content of the websites addressed by the links contained in e-mails. Data and information present in these websites, and the way in which the fraudulent websites ask the client for his/her

personal identification data and then process these data, clearly reveals the fraudulent or legitimate nature of the websites. Choosing a data mining method that extracts the regularities found in illegitimate websites leads the filtering system to discriminate between legitimate and illegitimate websites and so, between legitimate and illegitimate e-mais in a more accurate way. In (Castillo, 2007) a hybrid phishing filter consisting of the integrated application of three classifiers is shown. The first classifier deals with patterns extracted from textual content of the subject and body and it is based on a probabilistic paradigm. The second classifier based on rules analyses the non grammatical features contained in the body of e-mails (forms, executable code, and links). The third classifier focuses on the content of the websites addressed by the links contained in the body of e-mails. A finite state automata identifies the answers given by these websites when a fictitious user introduces false confidential data in them. The final decision of the filter about the e-mail nature integrates the local decisions of the three classifiers.

FUTURE TRENDS

Future research in the design and development of e-mail filters should be mainly focused on the following five points:

1. Finding an effective representation of attributes or features used in textual and non textual data contained in e-mails and in the websites addressed by links present in the body of e-mails.
2. The methods used to deal with these new data. It would be necessary to improve or adapt well known data mining methods for analysing the new information or to develop new classification methods suitable to the features of the new data.
3. Reducing the dependence between the classification performance and the size of the training sample used in the training phase of filters.
4. Learning capabilities. Many current filters are not endowed with the ability of learning from past mistakes because the chosen representation of the information does not allow evolution at all.
5. The integration of different methods of analysis and classification in order to design and develop hybrid systems that use different strategies cooperating to enhance the classification performance.

CONCLUSION

The classification performance of e-mail filters mainly depends on the representation chosen for the e-mail content and the methods working over the representation. Most current spam and phishing filters fail to obtain a good classification performance since the analysis and processing rely only on the e-mail source and the words in the body. The improvement of filters would imply to design and develop hybrid systems that integrate several sources of knowledge (textual content of emails, links contained in emails, addressed websites and heuristic knowledge) and different data mining techniques. A right representation and processing of all available kinds of knowledge will lead to optimize classification effectiveness and efficiency.

REFERENCES

Castillo, M.D., & Serrano, J.I. (2006). An Interactive Hybrid System for Identifying and Filtering Unsolicited E-mail. In E. Corchado, H. Yin, V. Botti and C. Fyfe (Eds.), *Lecture Notes in Computer Science: Vol. 4224* (pp. 779-788). Berlin: Springer-Verlag.

Castillo, M.D., Iglesias, A. & Serrano, J.I. (2007). An Integrated Approach to Filtering Phishing e-mails. In A. Quesada-Arencibia, J.C. Rodriguez, R. Moreno-Diaz jr, and R. Moreno-Diaz (Eds.), *11th International Conference on Computer Aided Systems Theory* (pp. 109-110). Spain: University of Las Palmas de Gran Canaria.

Communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee of the Regions on unsolicited commercial communications or 'spam'. (2004). Retrieved March 22, 2007, from http://www.icp.pt/txt/template16.jsp?categoryId=59469

Cunningham, P., Nowlan, N., Delany, S.J.& Haahr M. (2003) *A Case-Based Approach to Spam Filtering that Can Track Concept Drift*. (Tech. Rep. No. 16). Dublin: Trinity College, Department of Computer Science.

Daelemans, W., Zavrel, J., & van der Sloot, K. (2001) TiMBL: Tilburg Memory-Based Learner - version 4.0 Reference Guide. (Tech. Rep. No. 01-04). Netherlands: Tilburg University, Induction of Linguistic Knowledge Research Group.

GoDaddy (2006). Retrieved March 22, 2007, from Go Daddy Web Site: http://www.godaddy.com/

Good news: 'Phishing' scams net only $500 million. (2004). Retrieved March 22, 2007, from http://news.com.com/2100-1029_3-5388757.html

Green, T (2005). *Phishing—Past, present and future*. White Paper. Greenview Data Inc.

June Phishing Activity Trends Report. (2006). Retrieved October 16, 2006, from http://www.antiphishing.org

Mertz, D. (2002). *Spam Filtering Techniques: Six approaches to eliminating unwanted e-mail*. Retrieved March 22, 2007, from http://www-128.ibm.com/developerworks/linux/library/l-spamf.html

NetCraft (2004). Retrieved March 22, 2007, from Netcraft Web Site: http://news.netcraft.com

Pyzor (2002). Retrieved March 22, 2007, from Pyzor Web Site: http://pyzor.sourceforge.net

Randazzese, V. A.(2004). *ChoiceMail eases antispam software use while effectively fighting off unwanted e-mail traffic*. Retrieved March 22, 2007, from http://www.crn.com/security/23900678

Sebastiani, F. (2002). Machine Learning in Automated Text Categorization. *ACM Computing Surveys, 34*(1), 1-47.

Sergeant, M. (2003). *Internet-Level Spam Detection and SpamAssassin 2.50*. Paper presented at the 2003 Spam Conference, Cambridge, MA.

Serrano, J.I., & Castillo, M.D. (2007). Evolutionary Learning of Document Categories. *Journal of Information Retrieval, 10*, 69-83.

Spyware: Securing gateway and endpoint against data theft. (2007). White Paper. Sophos Labs.

Understanding Phishing and Pharming. (2006). White Paper. Retrieved March 22, 2007, from http://www.mcafee.com

Vinther, M. (2002). *Junk Detection using neural networks* (MeeSoft Technical Report). Retrieved March 22, 2007, from http://logicnet.dk/reports/JunkDetection/JunkDetection.htm

KEY TERMS

Classification Efficiency: Time and resources used to build a classifier or to classify a new e-mail.

Classification Effectiveness: Ability of a classifier to take the right classification decisions.

Data Mining: Application of algorithms for extracting patterns from data.

Hybrid System: System that processes data, information and knowledge taking advantage of the combination on individual data mining paradigms.

Integrated Approach: Approach that takes into account different sources of data, information and knowledge present in e-mails.

Phishing E-Mail: Unsolicited e-mail intending on stealing personal data for financial gain.

Spam E-Mail: Unsolicited commercial e-mail.

Data Mining for Structural Health Monitoring

Ramdev Kanapady
University of Minnesota, USA

Aleksandar Lazarevic
United Technologies Research Center, USA

INTRODUCTION

Structural health monitoring denotes the ability to collect data about critical engineering structural elements using various sensors and to detect and interpret adverse "changes" in a structure in order to reduce life-cycle costs and improve reliability. The process of implementing and maintaining a structural health monitoring system consists of operational evaluation, data processing, damage detection and life prediction of structures. This process involves the observation of a structure over a period of time using continuous or periodic monitoring of spaced measurements, the extraction of features from these measurements, and the analysis of these features to determine the current state of health of the system. Such health monitoring systems are common for bridge structures and many examples are citied in (Maalej et al., 2002).

The phenomenon of damage in structures includes localized softening or cracks in a certain neighborhood of a structural component due to high operational loads, or the presence of flaws due to manufacturing defects. Damage detection component of health monitoring system are useful for non-destructive evaluations that are typically employed in agile manufacturing systems for quality control and structures, such as turbine blades, suspension bridges, skyscrapers, aircraft structures, and various structures deployed in space for which structural integrity is of paramount concern (Figure 1). With the increasing demand for safety and reliability of aerospace, mechanical and civilian structures damage detection techniques become critical to reliable prediction of damage in these structural systems.

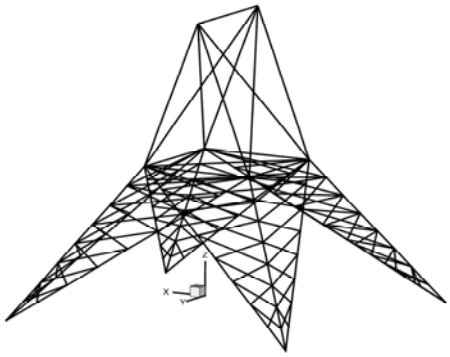
Most currently used damage detection methods are manual such as tap test, visual or specially localized measurement techniques (Doherty, 1997). These techniques require that the location of the damage have to be on the surface of the structure. In addition, location of the damage has to be known a priori and these locations have to be readily accessible. This makes current maintenance procedure of large structural systems very time consuming and expensive due to its heavy reliance on human labor.

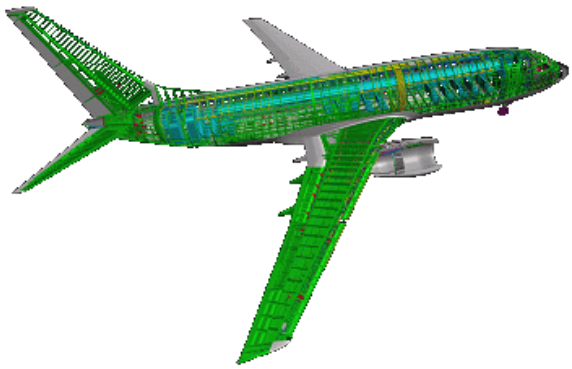
BACKGROUND

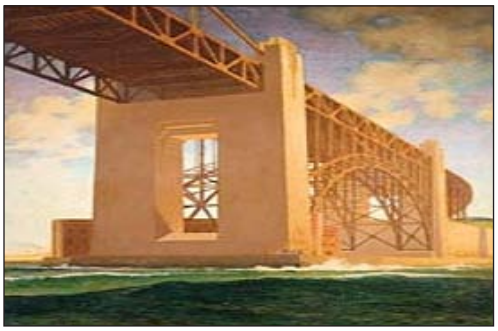
The damage in structures and structural systems is defined through comparison between two different states of the system, where the first one is the initial undamaged state, and the second one is damaged state. Emerging continuous monitoring of an instrumented structural system often results in the accumulation of a large amount of data that need to be processed, analyzed and interpreted for damage detection. However, the rate of accumulating such data sets far outstrips the ability to analyze them manually. As a result, there is a need to develop an intelligent data processing component that can significantly improve current damage detection systems. Since damage in changes in the properties of the structure or quantities derived from these properties, the process of health monitoring eventually reduces to a form of data mining problem. Design of data mining techniques that can enable efficient, real-time and robust (without false alarm) prediction of damage presents one of key challenging technological opportunity.

In recent years, various data mining techniques such as artificial neural networks (ANNs) (Anderson et al., 2003; Lazarevic et al., 2004; Ni et al., 2002; Sandhu et al., 2001; Yun & Bahng, 2000; Zhao, Ivan & DeWolf, 1998;), support vector machines (SVMs) (Mita & Hagiwara 2003), decision trees (Sandhu et al., 2001) have been successfully applied to structural damage detection problems. This success can be attributed to numerous disciplines integrated with data mining such as pattern recognition, machine learning and statistics. In addition, it is well known that data mining techniques can effectively handle noisy, partially incomplete and

Figure 1. Examples of engineering structures that require structural health monitoring systems







faulty data, which is particularly useful, since in damage detection applications, measured data are expected to be incomplete, noisy and corrupted.

The intent of this chapter is to provide a survey of emerging data mining techniques for structural health monitoring with particular emphasis on damage detection. Although the field of damage detection is very broad and consists of vast literature that is not based on data mining techniques, this survey will be predominantly focused on data-mining techniques for damage detection based on changes in properties of the structure.

CATEGORIZATION OF STRUCTURAL DAMAGE

The damage in structures can be classified as linear or nonlinear. Damage is considered as linear if the undamaged structure remains elastic after damage. However, if the initial structure behaves in a nonlinear manner after the damage initiation, then the damage is considered as nonlinear. However, it is possible that the damage is linear at the damage initiation phase, but after prolonged growth in time, it may become nonlinear. For example, loose connections between the structures at the joints or the joints that rattle (Sekhar, 2003) are considered non-linear damages. Examples of such non-linear damage detection systems are described in (Adams & Farrar, 2002; Kerschen & Golinval, 2004).

The most of the damage detection techniques in the literature are proposed for linear case. They are based on the following three levels of damage identification: 1. Recognition—qualitative indication that damage might be present in the structure, 2. Localization—information about the probable location of the damage in the structure, 3. Assessment—estimate of the extent of severity of the damage in the structure. Such damage detection techniques can be found in several approaches (Yun & Bahng, 2000; Ni et al., 2002; Lazarevic et al., 2004).

CLASSIFICATION OF DAMAGE DETECTION TECHNIQUES

We provide several different criteria for classification of damage detection techniques based on data mining.

In the first classification, damage detection techniques can be categorized into continuous (Keller & Ray, 2003) and periodic (Patsias & Staszewski, 2002) damage detection systems. While continuous techniques are mostly employed for real-time health monitoring, periodic techniques are employed for time-to-time health monitoring. Feature extraction from large amount of data collected from real time sensors can be a key distinction between these two techniques.

In the second classification, we distinguish between application-based and application-independent techniques. Application-based techniques are generally applicable to a specific structural system and they typically assume that the monitored structure responds in some predetermined manner that can be accurately modeled by (i) numerical techniques such as finite element (Sandhu et al., 2001) or boundary element analysis (Anderson et al., 2003) and/or (ii) behavior of the response of the structures that are based on physics based models (Keller & Ray, 2003). Most of damage detection techniques that exist in literature belong to application-based approach, where the minimization of the residue between the experimental and the analytical model is built into the system. Often, this type of data is not available and can render application-based methods impractical for certain applications particularly for structures that are designed and commissioned without such models. On the other hand, application-independent techniques do not depend on specific structure and they are generally applicable to any structural system. However, the literature on these techniques is very sparse and the research in this area is at very nascent stage (Bernal & Gunes, 2000; Zang et. al 2004).

In the third classification, damage detection techniques are split into signature based and non-signature based methods. Signature–based techniques extensively use signatures of known damages in the given structure that are provided by human experts. These techniques commonly fall into the category of recognition of damage detection, which only provides the qualitative indication that damage might be present in the structure (Friswell et al., 1994) and to certain extent the localization of the damage (Friswell et al., 1997). Non-signature methods are not based on signatures of known damages and they do not only recognize but also localize and assess extent of damage. Most of the damage detection techniques in the literature fall into this category (Lazarevic et al., 2004; Ni et al., 2002; Yun & Bahng 2000).

In the fourth classification, damage detection techniques are classified into local (Sekhar 2003; Wang, 2003) and global (Fritzen & Bohle 2001) techniques. Typically, the damage is initiated in a small region of the structure and hence it can be considered as local phenomenon. One could employ a local or global damage detection features that are derived from local or global response or properties of the structure. Although local features can detect the damage effectively, these features are very difficult to obtain from the experimental data such as higher natural frequencies and mode shapes of structure (Ni, Wang & Ko, 2002). In addition, since the vicinity of damage is not known a priori, the global methods that can employ only global damage detection features such lower natural frequencies of the structure (Lazarevic et al., 2004) are preferred.

Finally, damage detection techniques can be classified as traditional and emerging data-mining techniques. Traditional analytical techniques employ mathematical models to approximate the relationships between specific damage conditions and changes in the structural response or dynamic properties. Such relationships can be computed by solving a class of so-called inverse problems. The major drawbacks of the existing approaches are as follows: i) the more sophisticated methods involve computationally cumbersome system solvers which are typically solved by singular value decomposition techniques, non-negative least squares techniques, bounded variable least squares techniques, etc.; and, ii) all computationally intensive procedures need to be repeated for any newly available measured test data for a given structure. Brief survey of these methods can be found in (Doebling et al., 1996). On the other hand, data mining techniques consist of application techniques to model an explicit inverse relation between damage detection features and damage by minimization of the residue between the experimental and the analytical model at the training level. For example, the damage detection features could be natural frequencies, mode shapes, mode curvatures, etc. It should be noted that data mining techniques are also applied to detect features in large amount of measurement data. In the next few sections we will provide a short description of several types of data-mining algorithms used for damage detection.

Classification

Data mining techniques based on classification were successfully applied to identify the damage in the structures. For example, decision trees have been applied to detect damage in an electrical transmission tower (Sandhu et al., 2001). It has been found that in this approach decision trees can be easily understood, while many interesting rules about the structural damage were found. In addition, a method using SVMs has been proposed to detect local damages in a building structure (Mita & Hagiwara, 2003). The method is verified to have capability to identify not only the location of damage but also the magnitude of damage with satisfactory accuracy employing modal frequency patterns as damage features.

Pattern Recognition

Pattern recognition techniques are also applied to damage detection by various researchers. For example, the statistical pattern recognition is applied to damage detection employing relatively few measurements of modal data collected from three scale model-reinforced concrete bridges (Haritos & Owen, 2004), but the method could only be able to indicate that damage had occurred. In addition, independent component analysis, a multivariate statistical method also known as proper orthogonal decomposition, has been applied to damage detection problem on time history data to capture essential pattern of the measured vibration data (Zang et al., 2004).

Regression

Regression techniques such as radial basis functions (Önel et. al, 2006) and neutral networks (Lazarevic et al., 2004; Ni, Wang & Ko, 2002; Zhao et al., 1998) have been successfully applied to detect the existence, location and quantification of the damage in the structure employing modal data. Neural networks have been extremely popular in recent years due to their capabilities as universal approximators.

In damage detection approaches based on neural networks, the damage location and severity are simultaneously identified using one-stage scheme, as also called the direct method (Zhao et al., 1998), where the neural network is trained with different damage levels at each possible damage location. However, these studies were restricted to very small models with a small number of target variables (order of 10), and the development of a predictive model that could correctly identify the location and severity of damage in practical large-scale complex structures using this direct approach was a considerable challenge. Increased geometric complexity of the structure caused increase in the number of target variables thus resulting in data sets with large number of target variables. Since the number of prediction models that needs to be built for each continuous target variable increases, the number of training data records required for effective training of neural networks also increases, thus requiring more computational time for training neural networks but also more time for data generation, since each damage state (data record) requires an eigen solver to generate natural frequency and mode shapes of the structure. The earlier direct approach, employed by numerous researchers required the prediction of the material property, namely, the Young's modulus of elasticity considering all the elements in the domain individually or simultaneously. However, this approach does not scale to situations in which thousands of elements are present in the complex geometry of the structure or when multiple elements in the structure have been damaged simultaneously.

To reduce the size of the system under consideration, several substructure-based approaches have been proposed (Sandhu et al., 2002; Yun & Bahng, 2000). These approaches partition the structure into logical sub-structures, and then predict the existence of the damage in each of them. However, pinpointing the location of damage and extent of the damage is not resolved completely in these approaches. Recently, these issues have been addressed in two hierarchical approaches (Lazarevic et al., 2004; Ni et al., 2002). Ni (Ni et al., 2002) have proposed the approach where neural networks are hierarchically trained using one-level damage samples to first locate the position of the damage and then the network is re-trained by an incremental weight update method using additional samples corresponding to different damage degrees but only at the identified location at the first stage. The input attributes of neural networks are designed to depend only on damage location, and they consisted of several natural frequencies and a few incomplete modal vectors. Since measuring mode shapes are difficult, global method based only on natural frequencies are highly preferred. However, employing natural frequencies as

features traditionally has many drawbacks (e.g. two symmetric damage locations cannot be distinguished using only natural frequencies). To overcome these drawbacks, (Lazarevic et. al, 2004; Lazarevic et al., 2007) proposed an effective and scalable data mining technique based only on natural frequencies as features where symmetrical damage locations of damage, as well as spatial characteristics of structural systems are integrated in building the model. The proposed localized clustering-regression based approach consists of two phases: (*i*) finding the most similar training data records to a test data record of interest and creating a cluster from these data records; and (*ii*) predicting damage intensity only in those structure elements that correspond to data records from the built cluster, assuming that all other structure elements are undamaged. In the first step, for each test data record, they build a local cluster around that specific data record. The cluster contains the most similar data records from the training data set. Then, in the second step, for all identified similar training data records that belong to created cluster, they identify corresponding structure elements assuming that the failed element is one of these identified structure elements. By identifying corresponding structure elements the focus is only on prediction of the structure elements that are highly possible to be damaged. Therefore, instead of predicting an intensity of damage in all structure elements, a prediction model is built for each of these identified corresponding structure elements in order to determine which of these structure elements has really failed. Prediction model for a specific structure element is constructed using only those data records from the entire training data set that correspond to different intensities of its failure and using only a specific set of relevant natural frequencies. Experiments performed on the problem of damage prediction in a large electric transmission tower frame indicate that this hierarchical and localized clustering-regression based approach is more accurate than previously proposed hierarchical partitioning approaches as well as computationally more efficient.

Other Techniques

Other data mining based approaches have also been applied to different problems in structural health monitoring. For example, outlier based analysis techniques (Worden et al., 2000, Li, et al., 2006,) have been used to detect the existence of damage and wavelet based approaches (Wang, 2003; Bulut et. al., 2005) have been used to detect damage features or for data reduction component of structural heath monitoring system.

Hybrid Data Mining Models

For efficient, robust and scalable large-scale health monitoring system, that addresses all the three levels of damage detection, it is imperative data mining models include combination of classification, pattern recognition and/or regression type of methods. Combination of independent component analysis and artificial neural networks (Zang et al., 2004) has also been successfully applied to detect damages in structures. Recently, a hybrid local and global damage detection model employing SVM with the design of smart sensors (Hayano & Mita 2005) was also used. (Bulut et. al., 2005) presented a real-time prediction technique with wavelet based data reduction and SVM based classification of data at each sensor location and clustering based data reduction and SVM based classification at the global level for wireless network of sensors. For network of sensors architecture, an information fusion methodology that deals with data processing, feature extraction and damage decision techniques such as voting scheme, Bayesian inference, Dempster-Shafer rule and fuzzy inference integration in a hierarchical setup was recently presented (Wang et al. 2006).

FUTURE TRENDS

Damage detection is increasingly becoming an indispensable and integral component of any comprehensive structural health monitoring programs for mechanical and large-scale civilian, aerospace and space structures. Although a variety of techniques have been developed for detecting damages, there are still a number of research issues concerning the prediction performance and efficiency of the techniques that need to be addressed (Auwerarer & Peeters, 2003; De Boe & Golinval, 2001). In addition to providing three levels of damage identification, namely recognition, localization and assessment, future data mining research should also focus on providing another level of damage identification to include estimating the remaining service life of a structure.

CONCLUSION

In this chapter a survey of emerging data mining techniques for damage detection in structures is provided. This survey reveals that the existing data-mining techniques are predominantly based on changes in properties of the structure to classify, localize and predict the extent of damage.

REFERENCES

Adams, D., & Farrar, C. 2002. Classifying linear and nonlinear structural damage using frequency domain ARX models, *Structural Health Monitoring*, *1*(2), 185-201.

Anderson, T., Lemoine, G., & Ambur, D. (2003). An artificial neural network based damage detection scheme for electrically conductive composite structures, In *Proceedings of the 44th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Virginia.

Auwerarer, H. & Peeters, B. (2003). International research projects on structural health monitoring: An overview, *Structural Health Monitoring*, *2*(4), 341-358.

Bulut, A., Singh A. K., Shin, P., Fountain, T., Jasso, H., Yan, L., & Elgamal, A. (2005). Real-time nondestructive structural health monitoring using support vector machines and wavelets, In N. Meyendorf, G.Y. Baaklini, & B. Michel, (Eds.) *Proceedings of SPIE, Advanced Sensor Technologies for Nondestructive Evaluation and Structural Health Monitoring* (pp. 180-189).

Bernal, D. & Gunes, B. (2000). Extraction of system matrices from state-space realizations, In *Proceedings of the 14th Engineering Mechanics Conference*, Austin, TX.

Doherty, J. (1987). Nondestructive evaluation, In A.S. Kobayashi (Ed.) *Handbook on Experimental Mechanics*. Society of Experimental Mechanics, Inc.

De Boe, P. & Golinval, J.-C. (2001). Damage localization using principal component analysis of distributed sensor array, In F.K. Chang (Ed.) *Structural Health Monitoring*: *The Demands and Challenges* (pp. 860-861). Boca Raton FL: CRC Press.

Doebling, S., Farrar, C., Prime, M. & Shevitz, D. (1996). *Damage identification and health monitoring of structural systems from changes in their vibration characteristics: A literature review*, Report LA-12767-MS, Los Alamos National Laboratory.

Friswell, M., Penny J. & Wilson, D. (1994). Using vibration data and statistical measures to locate damage in structures, *Modal Analysis: The International Journal of Analytical and Experimental Modal Analysis*, *9*(4), 239-254.

Friswell, M., Penny J. & Garvey, S. (1997). Parameter subset selection in damage location, *Inverse Problems in Engineering*, *5*(3), 189-215.

Fritzen, C. & Bohle, K. (2001). Vibration based global damage identification: A tool for rapid evaluation of structural safety. In *Structural Health Monitoring*: The Demands and Challenges, Chang, F.K., Ed., CRC Press, Boca Raton, FL, 849-859.

Haritos, N. & Owen, J. S. (2004). The use of vibration data for damage detection in bridges: A comparison of system identification and pattern recognition approaches, *Structural health Monitoring*, *3*(2), 141-163.

Hayano, H. & Mita A. (2005), Structural health monitoring system using FBG sensor for simultaneous detection of acceleration and strain, In M. Tomizuka (Ed.) *Proceedings of SPIE, Smart Structures and Materials, Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems*, 5765 (pp. 624-633).

Kerschen, G. & Golinval, J-C. (2004). Feature extraction using auto-associative neural networks, *Smart Material and Structures*, *13*, 211-219.

Keller, E., & Ray, A. (2003). Real-time health monitoring of mechanical structures, *Structural Health Monitoring*, *2*(3), 191-203.

Khoo, L., Mantena, P., & Jadhav, P. (2004). Structural damage assessment using vibration modal analysis, *Structural Health Monitoring*, *3*(2), 177-194.

Lazarevic, A., Kanapady, R., Tamma, K. K., Kamath, C. & Kumar V. (2003). Localized prediction of continuous target variables using hierarchical clustering. In *Proceedings of the Third IEEE International Conference on Data Mining*, Florida.

Lazarevic, A., Kanapady, R., Tamma, K.K., Kamath, C., & Kumar, V. (2003). Damage prediction in structural mechanics using hierarchical localized clustering-based approach. *Data Mining and Knowledge Discovery: Theory, Tools, and Technology V*, Florida.

Lazarevic, A., Kanapady, R., Tamma, K. K., Kamath, C. & Kumar, V. (2004). Effective localized regression for damage detection in large complex mechanical structures. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Seattle.

Lazarevic, A., Kanapady, R., Kamath, C., Kumar, V. & Tamma, K. (2007). *Predicting very large number of target variables using hierarchical and localized clustering* (in review).

Li, H. C. H., Hersberg, I. & Mouritz, A. P. (2006). Automated characterization of structural disbonds by statistical examination of bond-line strain distribution, *Structural Health Monitoring*, *5*(1), 83-94.

Maalej, M., Karasaridis, A., Pantazopoulou, S. & Hatzinakos, D. (2002). Structural health monitoring of smart structures, *Smart Materials and Structures, 11*, 581-589.

Mita, A., & Hagiwara, H. (2003). Damage diagnosis of a building structure using support vector machine and modal frequency patterns, In *the Proceedings of SPIE* (Vol. 5057, pp. 118-125).

Ni, Y., Wang, B. & Ko, J. (2002). Constructing input vectors to neural networks for structural damage identification, *Smart Materials and Structures*, *11*, 825-833.

Önel, I., Dalci, B. K. & Senol, Ï., (2006), Detection of bearing defects in three-phase induction motors using Park's transform and radial basis function neural networks, *Sadhana*, *31*(3), 235-244.

Patsias, S. & Staszewski, W. J. (2002). Damage detection using optical measurements and wavelets, *Structural Heath Monitoring*, *1*(1), 5-22.

Sandhu, S., Kanapady, R., Tamma, K. K., Kamath, C., & V. Kumar. (2002). A Sub-structuring approach via data mining for damage prediction and estimation in complex structures, In *Proceedings of the SIAM International Conference on Data Mining*, Arlington, VA.

Sandhu, S., Kanapady, R., Tamma, K. K., Kamath, C. & Kumar, V. (2001). Damage prediction and estimation in structural mechanics based on data mining, In *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining/Fourth Workshop on Mining Scientific Datasets*, California.

Sekhar, S. (2003). Identification of a crack in rotor system using a model-based wavelet approach, *Structural Health Monitoring*, *2*(4), 293-308.

Wang, W. (2003). An evaluation of some emerging techniques for gear fault detection, *Structural Health Monitoring*, *2*(3), 225-242.

Wang, X., Foliente, G., Su, Z. & Ye, L. (2006). Multilevel decision fusion in a distributed active sensor network for structural damage detection, *Structural Health Monitoring*, *5*(1), 45-58.

Worden, K., Manson G., & Fieller N. (2000). Damage detection using outlier analysis, *Journal of Sound and Vibration*. *229*(3), 647-667.

Yun, C., & Bahng, E. Y. (2000). Sub-structural identification using neural networks, *Computers & Structures*, *77*, 41-52.

Zang, C., Friswell, M. I., & Imregun, M. (2004). Structural damage detection using independent component analysis, *Structural Health Monitoring*, *3*(1), 69-83.

Zhao. J., Ivan, J., & DeWolf, J. (1998). Structural damage detection using artificial neural networks, *Journal of Infrastructure Systems*, *4*(3), 93-101.

KEY TERMS

Damage: Localized softening or cracks in a certain neighborhood of a structural component due to high operational loads, or the presence of flaws due to manufacturing defects.

High Natural Frequency: Natural Frequency which measurement requires sophisticated instrumentation and it characterizes local response or properties of an engineering structure.

Low Natural Frequency: Natural Frequency that can be measured with simple instrumentation and it characterizes global response or properties of an engineering structure.

Modal Properties: Natural frequency, mode shapes and mode curvatures constitutes modal properties.

Mode Shapes: Eigen vectors associated with the natural frequencies of the structure.

Natural Frequency: Eigen values of the mass and stiffness matrix system of the structure.

Smart Structure: A structure with a structurally integrated fiber optic sensing system.

Structures and Structural System: The word structure used has been employed loosely in the chapter. The structure is referred to the continuum material and whereas the structural system consists of structures which are connected at joints.

Data Mining for the Chemical Process Industry

Ng Yew Seng
National University of Singapore, Singapore

Rajagopalan Srinivasan
National University of Singapore and Institute of Chemical & Engineering Sciences, Singapore

INTRODUCTION

Advancements in sensors and database technologies have resulted in the collection of huge amounts of process data from chemical plants. A number of process quantities such as temperature, pressure, flow rates, level, composition, and pH can be easily measured. Chemical processes are dynamic systems and are equipped with hundreds or thousands of sensors that generate readings at regular intervals (typically seconds). In addition, derived quantities that are functions of the sensor measurements as well as alerts and alarms are generated regularly. Several commercial data warehouses, referred to as plant historians in chemical plants, such as the DeltaV Continuous Historian (from Emerson Process Management), InfoPlus.21™ (from AspenTech), Uniformance® PHD (from Honeywell), and Industrial SQL (from Wonderware) are in common use today around the world. These historians store large amount (weeks) of historical process operation data at their original resolution and an almost limitless amount (years) in compressed form. This data is available for mining, analysis and decision support – both real-time and offline.

Process measurements can be classified based on their nature as binary (on/off) or continuous. However, both are stored in discrete form in the historians. Measurements can also be classified based on their role during operation as controlled, manipulated, and non-control related variables. Controlled variables are directly or indirectly related to the plant's quality, production, or safety objectives and are maintained at specified setpoints, even in the face of disturbances, by analog or digital controllers. This regulation is achieved by altering manipulated variables such as flow-rates. Chemical plants are typically well-integrated – a change in one variable would propagate across many others. Non-control related variables do not have any role in plant control, but provide information to plant personnel regarding the state of the process.

In general, a plant can operate in a number of states which can be broadly classified into steady-states and transitions (Srinivasan *et al.*, 2005b). Large scale plants such as refineries typically run for long periods in steady-states but undergo transition if there is a change in feedstock or product grades. Transitions also result due to large process disturbances, maintenance activities, and abnormal events. During steady-states, the process variables vary within a narrow range. In contrast, transitions correspond to large changes / discontinuities in the plant operations; *i.e.*, change of set points, turning on or idling of equipments, valve manipulations, etc. A number of decisions are needed on the part of the plant personnel to keep the plant running safely and efficiently during steady states as well as transitions. Data mining and analysis tools that facilitate humans to uncover information, knowledge, patterns, trends, and relationships from the historical data are therefore crucial.

BACKGROUND

Numerous challenges bedevil the mining of data generated by chemical processes. These arise from the following general characteristics of the data:

1. *Temporal*: Since the chemical process is a dynamic system, all measurements vary with time.
2. *Noisy*: The sensors and therefore the resulting measurements can be significantly noisy.
3. *Non-stationarity*: Process dynamics can change significantly, especially across states because of structural changes to the process. Statistical properties of the data such as mean and variance can therefore change significantly between states.
4. *Multiple time-scales*: Many processes display multiple time scales with some variables varying quickly (order of seconds) while others respond over hours.

5. *Multi-rate sampling*: Different measurements are often sampled at different rates. For instance, online measurements are often sampled frequently (typically seconds) while lab measurements are sampled at a much lower frequency (a few times a day).
6. *Nonlinearity*: The data from chemical processes often display significant nonlinearity.
7. *Discontinuity:* Discontinuous behaviors occur typically during transitions when variables change status – for instance from inactive to active or no flow to flow.
8. *Run-to-run variations*: Multiple instances of the same action or operation carried out by different operators and at different times would not match. So, signals from two instances could be significantly different due to variation in initial conditions, impurity profiles, exogenous environmental or process factors. This could result in deviations in final product quality especially in batch operations (such as in pharmaceutical manufacturing).

Due to the above, special purpose approaches to analyze chemical process data are necessary. In this chapter, we review these data mining approaches.

MAIN FOCUS

Given that large amounts of operational data are readily available from the plant historian, data mining can be used to extract knowledge and improve process understanding – both in an offline and online sense. There are two key areas where data mining techniques can facilitate knowledge extraction from plant historians, namely (i) process visualization and state-identification, and (ii) modeling of chemical processes for process control and supervision.

Visualization techniques use graphical representation to improve human's understanding of the structure in the data. These techniques convert data from a numeric form into a graphic form that facilitates human understanding by means of the visual perception system. This enables post-mortem analysis of operations towards improving process understanding or developing process models or online decision support systems. A key element in data visualization is dimensionality reduction. Subspace approaches such as principal components analysis and self-organizing maps have been popular for visualizing large, multivariate process data. For instance, an important application is to identify and segregate the various operating regimes of a plant. Figure 1 shows the various periods over a course of two weeks when a refinery hydrocracker was operated in steady-states or underwent transitions. Such information is necessary to derive operational statistics of the operation. In this example, the states have been identified using two principal components that summarize the information in over 50 measurements. As another illustration, oscillation of process variables during steady-state operations is common but undesirable. Visualization techniques and dimensionality reduction can be applied to detect oscillations (Thornhill and Hagglund, 1997). Other applications include process automation and control (Amirthalingam *et al*., 2000; Samad *et al.*, 2007), inferential sensing (Fortuna *et al.*, 2005), alarm management (Srinivasan *et al.*, 2004a), control loop performance analysis (Huang, 2003) and preventive maintenance (Harding *et al*., 2007).

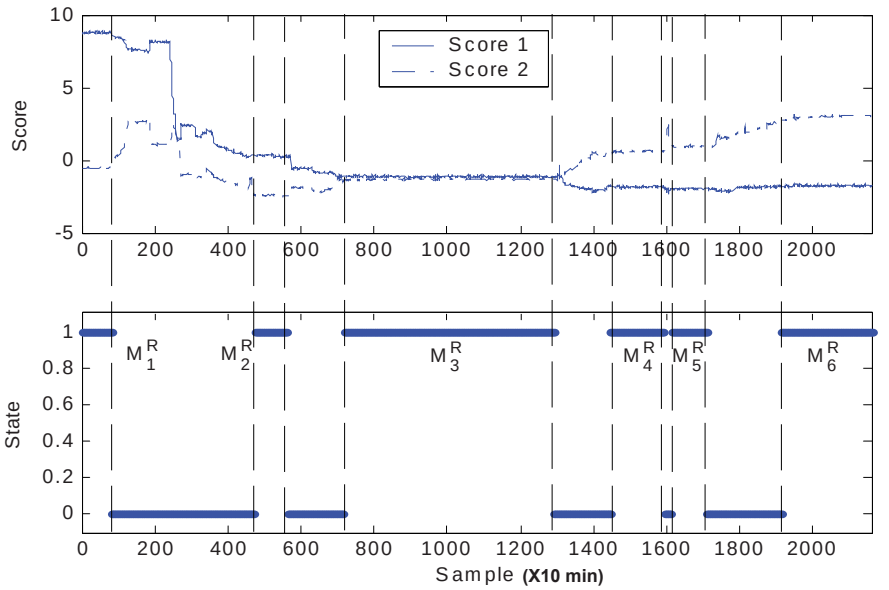
Data-based models are also frequently used for process supervision – fault detection and identification (FDI). The objective of FDI is to decide in real-time the condition – normal or abnormal – of the process or its constituent equipment, and (ii) in case of abnormality, identify the root cause of the abnormal situation. It has been reported that approximately 20 billion dollars are lost on an annual basis by the US petrochemical industries due to inadequate management of abnormal situations (Nimmo, 1995). Efficient data mining algorithms are hence necessary to prevent abnormal events and accidents. In the chemical industry, pattern recognition and data classification techniques have been the popular approaches for FDI. When a fault occurs, process variables vary from their nominal ranges and exhibit patterns that are characteristic of the fault. If the patterns observed online can be matched with known abnormal patterns stored in a database, the root cause of a fault can generally be identified.

In the following, we review popular data mining techniques by grouping them into machine-learning, statistical, and signal processing approaches.

Machine Learning Methods

Neural-network (NN) is a popular machine learning technique that exhibits powerful classification and func-

Figure 1. Classification of process steady-states and transitions



tion approximation capability. It has been shown to be robust to noise and able to model nonlinear behaviors. It has hence become popular for process control, monitoring, and fault diagnosis. Reviews of the applications of neural-networks in the chemical process industry have been presented in Hussain (1999) and Himmelblau (2000). Although in theory, artificial neural-networks can approximate any well-defined non-linear function with arbitrary accuracy, the construction of an accurate neural classifier for complex multivariate, multi-class, temporal classification problems such as monitoring transient states suffers from the "curse of dimensionality". To overcome this, new neural-network architectures such as one-variable-one-network (OVON) and one-class-one-network (OCON) have been proposed by Srinivasan *et al.* (2005a). These decompose the original classification problem into a number of simpler classification problems. Comparisons with traditional neural networks indicate that the new architectures are simpler in structure, faster to train, and yield substantial improvement in classification accuracy.

The Self-Organizing Map (SOM) is another type of neural-network based on unsupervised learning. It is capable of projecting high-dimensional data to a two dimensional grid, and hence provides good visualization facility to data miners. Ng *et al.* (2006) demonstrated that data from steady-state operations are reflected as clusters in the SOM grid while those from transitions are shown as trajectories. Figure 2 depicts four different transitions that were commonly carried out in the boiler unit in a refinery. The SOM can therefore serve as a map of process operations and can be used to visually identify the current state of the process.

Classical PCA approaches cannot be directly applied to transitions data because they do not satisfy the statistical stationarity condition. An extension of PCA called Multiway-PCA was proposed by Nomikos and MacGregor (1994) that reorganizes the data into time-ordered blocks. A time-based scaling is applied to different samples based on the nominal operation of the transition. This leads to a normalized variable that is stationary and can be analyzed by PCA. However, this requires every run of data to have equal length, and each time point to be synchronized. An alternate approach is the so called dynamic PCA (DPCA) where each time instant is modeled using the samples at that point as well as additional time-lagged ones (Ku *et al.*, 1995). Srinivasan *et. al.* (2004b) used DPCA to classify transitions as well as steady states. Multi-model approaches, where different PCA models are developed for different states, offer another way to model such complex operations (Lu and Gao, 2005; Doan and Srinivasan, 2008). The extent of resemblance between the different models can be compared using PCA similarity factors. Comparison

of different instances of the same transition can also be performed to extract best practices (Ng *et al.*, 2006).

Signal Processing Methods

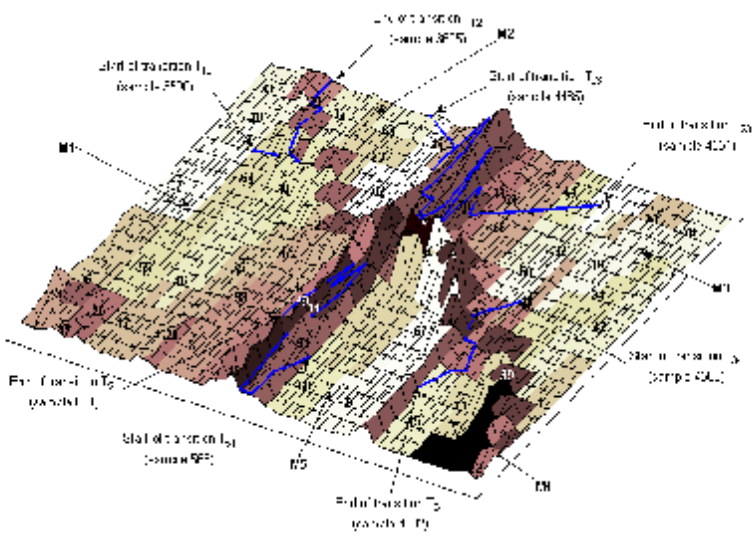
The above data mining methods abstract information from the data in the form of a model. Signal comparison methods on the other hand rely on a direct comparison of two instances of signals. Signal comparison methods thus place the least constraint on the characteristics of the data that can be processed, however they are computationally more expensive. The comparison can be done at a low level of resolution – comprising process trends – or at a high level of resolution, i.e., the sensor measurements themselves. Trend analysis is based on the granular representation of process data through a set of qualitative primitives (Venkatasubramanian *et al.*, 2003), which qualitatively describe the profile of the evolution. Different process states – normal and abnormal, map to different sequences of primitives. So a characteristic trend signature can be developed for each state, hence allowing a fault to be identified by template matching. A review of trend analysis in process fault diagnosis can be found in Maurya *et al.* (2007), where various approaches for online trend extraction and comparison are discussed.

Signal comparison at the process measurement level has been popular when non-stationary data, with different run lengths, have to be compared. Time synchronization between the signals is required in such cases. Dynamic time warping (DTW) has been frequently used for this purpose (Kassidas *et al.*,1998). As DTW is computationally expensive for long, multi-variate signals, several simplifications have been proposed. One such simplification relies on landmarks in the signal (such as peaks or local minima) to decompose a long signal into multiple shorter segments which are in turn compared using time warping (Srinivasan and Qian, 2005; 2007). However, one key requirement of DTW is that the starting and ending points of the signals to be compared should coincide. This obviates its direct use for online applications since the points in the historical database that should be matched with the starting and ending points of the online signal are usually unknown. A technique called dynamic locus analysis, based on dynamic programming, has been recently proposed for such situations (Srinivasan and Qian, 2006).

FUTURE TRENDS

Data analysis tools are necessary to aid humans visualize high dimensional data, analyze them, and extract actionable information to support process operations. A variety of techniques with differing strengths and weaknesses have been developed to date. The ideal data mining method would (i) adapt to changing operating and environmental conditions, and be tolerant to noise and uncertainties over a wide range, (ii) provide

Figure 2. Visualization of multi-state, multivariate temporal data with SOM



explanation on the patterns observed, their origin and propagation among the variables, (iii) simultaneously analyze many process variables, (iv) detect and identify novel patterns, (v) provide fast processing with low computational requirement, and (vi) be easy to implement and maintain. As shown in Table 1, no one technique has all these features necessary for large-scale problems (Srinivasan, 2007). Additional insights from human knowledge of process principles and causality can resolve some of the uncertainties that mere data cannot. Since no single method can completely satisfy all the important requirements, effective integration of different data mining methods is necessary. Multi-agent approaches (Wooldridge, 2002) could pave the way for such integration (Davidson *et al.*, 2006) and collaboration between the methods (Ng and Srinivasan, 2007). When multiple methods are used in parallel, a conflict resolution strategy is needed to arbitrate among the contradictory decisions generated by the various methods. Decision fusion techniques such as voting, Bayesian, and Dempster-Shafer combination strategies hold much promise for such fusion (Bossé *et al.*, 2006).

CONCLUSION

Data in the process industry exhibit characteristics that are distinct from other domains. They contain rich information that can be extracted and used for various purposes. Therefore, specialized data mining tools that account for the process data characteristics are required. Visualization and clustering tools can uncover good operating practices that can be used to improve plant operational performance. Data analysis methods for FDI can reduce the occurrence of abnormal events and occupational injuries from accidents. Collaborative multi-agent based methodology holds promise to integrate the strengths of different data analysis methods and provide an integrated solution.

REFERENCES

Amirthalingam, R., Sung, S.W., Lee, J.H., (2000). Two-step procedure for data-based modeling for inferential control applications, *AIChE Journal*, 46(10), Oct, 1974-1988.

Bossé, É, Valin, P., Boury-Brisset, A-C., Grenier, D., (2006). Exploitation of a priori knowledge for information fusion, *Information Fusion*, 7, 161-175.

Chiang, L.H. & Braatz, R.D., (2003). Process monitoring using causal map and multivariate statistics: fault detection and identification, *Chemometrics and Intelligent Laboratory Systems*, 65, 159 – 178.

Cho, J-H., Lee, J-M., Choi, S.W., Lee, D., Lee, I-B., (2005). Fault identification for process monitoring using kernel principal component analysis, *Chemical Engineering Science*, 60(1), Jan, 279-88.

Davidson, E.M., McArthur, S.D.J., McDonald, J.R., Cumming, T., Watt, I., (2006). Applying multi-agent system technology in practice: automated management and analysis of SCADA and digital fault recorder data, *IEEE Transactions on Power Systems*, 21(2), May, 559-67.

Doan, X-T, & Srinivasan, R. (2008). Online monitoring of multi-phase batch processes using phase-based multivariate statistical process control, *Computers and Chemical Engineering*, 32(1-2), 230-243.

Table 1. Strengths and shortcomings of different data mining methods

Methods	Knowledge-based	Machine Learning	Statistical Methods	Signal Processing
i) Adaptation & Robustness	×	×	√	√
ii) Explanation Facility	√	×	×	×
iii) Multivariate Analysis	×	√	√	×
iv) Novel Pattern Detection	×	√	√	√
v) Computational Complexity	√	√	√	×
vi) Ease of Development	×	√	√	√

Fortuna, L., Graziani, S., Xibilia, M.G. (2005). Soft sensors for product quality monitoring in debutanizer distillation columns, *Control Engineering Practice*, 13, 499-508.

Harding, J.A., Shahbaz, M., Srinivas, S., Kusiak, A., (2007). Data mining in manufacturing: A review, *Journal of Manufacturing Science and Engineering,* 128(4), 969-976.

Himmelblau, D.M., (2000). Applications of artificial neural networks in chemical engineering, *Korean Journal of Chemical Engineering*, 17(4), 373-392.

Huang, B., (2003). A pragmatic approach towards assessment of control loop performance, *International Journal of Adaptive Control and Signal Processing*, 17(7-9), Sept-Nov, 589-608.

Hussain, M.A., (1999). Review of the applications of neural networks in chemical process control – simulation and online implementation, *Artificial Intelligence in Engineering*, 13, 55-68.

Kassidas, A., MacGregor, J.F., Taylor, P.A., (1998). Synchronization of batch trajectories using dynamic time warping, *American Institute of Chemical Engineers Journal*, 44(4), 864–875.

Kourti, T., (2002). Process analysis and abnormal situation detection: From theory to practice, *IEEE Control System Magazine*, 22(5), Oct., 10-25.

Ku, W., Storer, R.H., Georgakis, C., (1995). Disturbance detection and isolation by dynamic principal component analysis, *Chemometrics and Intelligent Laboratory System*, 30, 179-196.

Lu, N., & Gao, F., (2005). Stage-based process analysis and quality prediction for batch processes, *Industrial and Engineering Chemistry Research*, 44, 3547-3555.

Maurya, M.R., Rengaswamy, R., Venkatasubramanian, V., (2007). Fault diagnosis using dynamic trend analysis: A review and recent developments, *Engineering Applications of Artificial Intelligence*, 20, 133-146.

Muller, K-R., Mika, S., Ratsch, G., Tsuda, K., Scholkopf, B., (2001). An introduction to kernel-based learning algorithms, *IEEE Transactions on Neural Networks*, 12(2), Mar, 181-201.

Ng, Y.S., & Srinivasan, R., (2007). Multi-agent framework for fault detection & diagnosis in transient operations, *European Symposium on Computer Aided Process Engineering (ESCAPE 27)*, May 27-31, Romania.

Ng, Y.S., Yu, W., Srinivasan, R., (2006). Transition classification and performance analysis: A study on industrial hydro-cracker, *International Conference on Industrial Technology ICIT 2006*, Dec 15-17, Mumbai, India, 1338-1343.

Nimmo, I., (1995). Adequately address abnormal operations, *Chemical Engineering Progress*, September 1995.

Nomikos, P., & MacGregor, J.F., (1994). Monitoring batch processes using multiway principal component analysis, *AIChE Journal, 40*, 1361-1375.

Qin, S. J., (2003). Statistical process monitoring: basics and beyond, *Journal of Chemometrics,* 17, 480-502.

Srinivasan, R., (2007). Artificial intelligence methodologies for agile refining: An overview, *Knowledge and Information Systems Journal*, 12(2), 129-145.

Srinivasan, R., & Qian, M., (2005). Offline temporal signal comparison using singular points augmented time warping, *Industrial & Engineering Chemistry Research*, 44, 4697-4716.

Srinivasan, R., & Qian, M., (2006). Online fault diagnosis and state identification during process transitions using dynamic locus analysis, *Chemical Engineering Science*, 61(18), 6109-6132.

Srinivasan, R., & Qian, M., (2007). Online temporal signal comparison using singular points augmented time warping, *Industrial & Engineering Chemistry Research*, 46(13), 4531-4548.

Srinivasan, R., Liu, J., Lim, K.W., Tan, K.C., Ho, W.K., (2004a). Intelligent alarm management in a petroleum refinery, *Hydrocarbon Processing*, Nov, 47–53.

Srinivasan, R., Wang, C., Ho, W.K., Lim, K.W., (2004b). Dynamic principal component analysis based methodology for clustering process states in agile chemical plants, *Industrial and Engineering Chemistry Research*, 43, 2123-2139.

Srinivasan, R., Wang. C., Ho, W.K., Lim, K.W., (2005a). Neural network systems for multi-dimensional temporal pattern classification, *Computers & Chemical Engineering*, 29, 965-981.

Srinivasan, R., Viswanathan, P., Vedam, H., Nochur, A., (2005b). A framework for managing transitions in chemical plants, *Computers & Chemical Engineering*, 29(2), 305-322

Samad, T., McLaughlin, P., Lu, J., (2007). System architecture for process automation: Review and trends, *Journal of Process Control*, 17, 191-201.

Thornhill, N.F., & Hagglund, T., (1997). Detection and diagnosis of oscillation in control loops, *Control Engineering Practice*, 5(10), Oct, 1343-54.

Wise, B.M., Ricker, N.L., Veltkamp, D.J., Kowalski, B.R., (1990). A theoretical basis for the use of principal components model for monitoring multivariate processes, *Process Control and Quality*, 1(1), 41-51.

Wooldridge, M., (2002). An introduction to multiagent systems, *John Wiley & Sons, Ltd*, West Sussex, England.

Venkatasubramanian, V., Rengaswamy, R., Kavuri, S.N., Yin, K., (2003). A review of process fault detection and diagnosis Part III: Process history based methods, *Computers and Chemical Engineering*, 27, 327 – 346.

KEY TERMS

Data Mining: The process of discovering meaningful information, correlations, patterns, and trends from databases, usually through visualization or pattern recognition techniques.

Fault Detection and Identification: The task of determining the health of a process (either normal or abnormal) and locating the root cause of any abnormal behavior.

Plant Historian: The data warehouse in a chemical plant. They can store large number of process variables such as temperature, pressure, and flow rates, as well as a history of the process operation, such as equipment settings, operator logs, alarms and alerts.

Process Industry: An industry in which raw materials are mixed, treated, reacted, or separated in a series of stages using chemical process operations. Examples of process industries include oil refining, petrochemicals, pharmaceuticals, water and sewage treatment, food processing, etc.

Process Supervision: The task of monitoring and overseeing the operation of a process. It involves monitoring of the process as well as of the activities carried out by operators.

Root Cause: The main reason for any variation in the observed patterns from an acceptable set corresponding to normal operations

Visualization: A technique that uses graphical representation to improve human's understanding of patterns in data. It converts data from a numeric form into a graphic that facilitates human understanding by means of the visual perception system.

Data Mining in Genome Wide Association Studies

Tom Burr
Los Alamos National Laboratory, USA

INTRODUCTION

The genetic basis for some human diseases, in which one or a few genome regions increase the probability of acquiring the disease, is fairly well understood. For example, the risk for cystic fibrosis is linked to particular genomic regions. Identifying the genetic basis of more common diseases such as diabetes has proven to be more difficult, because many genome regions apparently are involved, and genetic effects are thought to depend in unknown ways on other factors, called covariates, such as diet and other environmental factors (Goldstein and Cavalleri, 2005).

Genome-wide association studies (GWAS) aim to discover the genetic basis for a given disease. The main goal in a GWAS is to identify genetic variants, single nucleotide polymorphisms (SNPs) in particular, that show association with the phenotype, such as "disease present" or "disease absent" either because they are causal, or more likely, because they are statistically correlated with an unobserved causal variant (Goldstein and Cavalleri, 2005). A GWAS can analyze "by DNA site" or "by multiple DNA sites. " In either case, data mining tools (Tachmazidou, Verzilli, and De Lorio, 2007) are proving to be quite useful for understanding the genetic causes for common diseases.

BACKGROUND

A GWAS involves genotyping many cases (typically 1000 or more) and controls (also 1000 or more) at a large number (10^4 to 10^6) of markers throughout the genome. These markers are usually SNPs. A SNP occurs at a DNA site if more than one nucleotide (A, C, T, or G) is found within the population of interest, which includes the cases (which have the disease being studied) and controls (which do not have the disease). For example, suppose the sequenced DNA fragment from subject 1 is AAGC**C**TA and from subject 2 is AAGC**T**TA. These contain a difference in a single nucleotide. In this case there are two alleles ("alleles" are variations of the DNA in this case), C and T. Almost all common SNPs have only two alleles, often with one allele being rare and the other allele being common.

Assume that measuring the DNA at millions of sites for thousands of individuals is feasible. The resulting measurements for n_1 cases and n_2 controls are partially listed below, using arbitrary labels of the sites such as shown below. Note that DNA site 3 is a candidate for an association, with T being the most prevalent state for cases and G being the most prevalent state for controls.

	123	456	789 ...
Case 1:	AA**T**	CT**A**	TAT ...
Case 2:	A* **T**	CT**C**	TAT …
...			
Case n_1:	AA**T**	CT**G**	TAT ...
Control 1:	AA**G**	CT**A**	TTA ...
Control 2:	AA**G**	CT**A**	TTA ...
...			
Control n_2:	AA**G**	CT**A**	TTA ...

Site 6 is also a candidate for an association, with state A among the controls and considerable variation among the cases. The * character (case 2) can denote missing data, an alignment character due to a deletion mutation, or an insertion mutation, etc. (Toivonen et al., 2000).

In this example, the eye can detect such association candidates "by DNA site." However, suppose the collection of sites were larger and all n_1 cases and n_2 controls were listed, or that the analysis were "by haplotype." In principle, the haplotype (one "half" of the genome of a paired-chromosome species such as humans) is the entire set of all DNA sites in the entire genome. In practice, haplotype refers to the sequenced sites, such as those in a haplotype mapping (HapMap,

2005) involving SNPs as we focus on here. Both a large "by DNA site" analysis and a haplotype analysis, which considers the joint behavior of multiple DNA sites, are tasks that are beyond the eye's capability.

Using modern sequencing methods, time and budget constraints prohibit sequencing all DNA sites for many subjects (Goldstein and Cavalleri, 2005). Instead, a promising shortcut involves identifying haplotype blocks (Zhang and Jin, 2003; Zhang et al., 2002). A haplotype block is a homogeneous region of DNA sites that exhibit high linkage disequilibrium. Linkage disequilibrium between two DNA sites means there is negligible recombination during reproduction, thus "linking" the allelic states far more frequently than if the sites evolved independently. The human genome contains regions of very high recombination rates and regions of very low recombination rates (within a haplotype block). If a haplotype block consists of approximately 10 sites, then a single SNP marker can indicate the DNA state (A, C, T, or G) for each site in the entire block for each subject, thus reducing the number of sequenced sites by a factor of 10.

The HapMap project (HapMap, 2005) has led to an increase from approximately 2 million known SNPs to more than 8 million. Many studies have reported low haplotype diversity with a few common haplotypes capturing most of the genetic variation. These haplotypes can be represented by a small number of haplotype-tagging SNPs (htSNPs). The presence of haplotype blocks makes a GWAS appealing, and summarizes the distribution of genetic variation throughout the genome. SNPs are effective genetic markers because of their abundance, relatively low mutation rate, functional relevance, ease of automating sequencing, and role as htSNPs. The HapMap project is exploiting the concept that if an htSNP correlates with phenotype, then some of the SNPs in its "association block" are likely to be causally linked to phenotype.

MAIN THRUST

Data Mining

Data mining involves the extraction of potentially useful information from data. Identifying genomic regions related to phenotype falls within the scope of data mining; we will limit discussion to a few specific data mining activities, which can all be illustrated using the following example. Consider the 10 haplotypes (rows) below (Tachmazidou et al., 2007) at each of 12 SNPs (columns). The rare allele is denoted "1," and the common allele is denoted "0." By inspection of the pattern of 0s and 1s, haplotypes 1 to 4 are somewhat distinguishable from haplotypes 5 to 10. Multidimensional scaling (Figure 1) is a method to display the distances between the 45 pairs of haplotypes (Venables and Ripley, 1999). Although there are several evolutionary-model-based distance definitions, the Manhattan distance (the number of differences between a given pair of haplotype) is defensible, and was used to create all three plots in Figure 1. The top plot in Figure 1 suggests that there are two or three genetic groups. Ideally, if there are only two phenotypes (disease present or absent), then there would be two genetic groups that correspond to the two phenotypes. In practice, because common diseases are proving to have a complex genetic component, it is common to have more than two genetic groups, arising, for example, due to racial or geographic subdivision structures in the sampled population (Liu et al., 2004).

haplotype1	1	0	0	0	1	0	1	0	0	0	1	0
haplotype2	1	0	0	0	1	0	1	0	0	0	0	0
haplotype3	1	0	0	0	0	0	1	0	0	0	0	0
haplotype4	1	0	0	0	0	0	0	0	0	1	0	0
haplotype5	0	1	0	0	0	1	0	0	0	0	0	1
haplotype6	0	1	0	0	0	1	0	0	0	0	0	0
haplotype7	0	0	0	1	0	1	0	0	0	0	0	0
haplotype8	0	0	0	1	0	1	0	0	1	0	0	0
haplotype9	0	0	0	1	0	1	0	1	1	0	0	0
haplotype10	0	0	1	0	0	1	0	0	0	0	0	0

The data mining activities described below include: defining genetics-model-based distance measures; selecting features; control of the false alarm rate; clustering in the context of phylogenetic tree building, and genomic control using genetic model fitting to protect against spurious association between halplotype and disease status.

Defining Genetics-Model-Based Distance Measures

Effective haplotype blocks in a GWAS requires small "within-block" variation relative to "between-block" variation. Variation can be defined and measured in several ways. One way is the Manhattan distance be-

tween pairs of corresponding haplotype blocks from two subjects. In cases, a more appropriate distance measure can accommodate missing data, insertion mutations, deletion mutations. and weights DNA differences by an evolutionary model under which, for example, an A to G mutation ("transition") is more likely than an A to T or A to C ("transversions") (Burr, 2006; Burr et al., 2002). As with most distance-based analyses, the choice of distance measure can impact the conclusions. A reasonable approach to choosing a distance measure is to select the most defensible evolutionary model for the genome region under study and then choose the corresponding distance measure. The GWAS concept is then to search for excess similarity among haplotypes from affected individuals (the "disease present" phenotype).

Feature Selection

One-SNP-site-at-a-time screening as previously illustrated is common, and often uses a statistical measure of association between allele frequency and phenotype (Stram, 2004). One such measure is Pearson's χ^2 test statistic that tests for independence between phenotype and genotype. This leads to an efficient preliminary screening for candidate association sites by requiring the minimum allele frequency to be above a threshold that depends on the false alarm rate (Hannu et al., 2000). This screening rule immediately rejects most candidate SNPs, thus greatly reducing the number of required calculations of the χ^2 statistic.

Defining and finding haplotype blocks is a data mining activity for which several methods have been proposed. Finding effective blocks is a specialized form of "feature selection," which is a common data mining activity. The task with the GWAS context is to find partition points and associated blocks such that the blocks are as large and as homogeneous as possible, as defined by block size and homogeneity specifications. Liu et al. (2004) describe a dynamic programming method to find effective blocks and show that the block structure within each of 16 particular human subpopulations is considerably different, as is the block structure for the entire set of 16 subpopulations. This implies that most if not all subpopulations will need to be deliberately included for an effective GWAS. A second feature selection approach (Stram, 2004) is based on choosing htSNPs in order to optimize the predictability of unmeasured SNPs. The concept is that the haplotype should be increasingly well predicted as more SNPs are genotyped, and leads to a different search strategy than the ones described by Liu et al. (2004).

A third feature selection approach invokes the well-known scan statistic. There are many versions of the scan statistic, but any scan statistic (Sun et al., 2006) scans a "window" of fixed or variable length to find regions of interest, based for example on elevated count rates within the window, or in the GWAS context, based on regions showing a strong clustering of low "*p*"-values associated with large values of the χ^2 statistic. The "*p*" value in this context is the probability of observing a χ^2 statistic as large as or larger than what is actually observed in the real (phenotype, genotype) data.

Controlling the False Alarm Rate

As with many data mining analyses, the false alarm rate in searching for SNPs that are associated with phenotype can be quite high; however, suitable precautions such as a permutation test reference distribution are straightforward, although they are computationally intensive (Toivonen et al., 2000). The permutation test assumes the hypothesis "no association between the SNP and phenotype," which would imply that the "case" and "control" labels are meaningless labels that can be randomly assigned (permuted). This type of random reassignment is computed many times, and in each random reassignment, the test statistic such as the χ^2 test is recalculated, producing a reference distribution of χ^2 values that can be compared with the χ^2 value computed using the actual case and control labels.

Clustering

Clustering involves partitioning sampled units (haplotypes in our case) into groups and is one of the most commonly-used data mining tools (Tachmazidou et al., 2007). Finding effective ways to choose the number of groups and the group memberships is a large topic in itself. Consider the aforementioned example with 10 subjects, each measured at 12 SNP sites. The top plot in Figure 1 is a two-dimensional scaling plot (Venables and Ripley, 1999) of the matrix of pairwise Manhattan distances among the 10 example haplotypes. Haplotypes {1,2,3,4}, {5,6,10}, and {7,8,9} are candidate clusters. Methods to choose the number of clusters are beyond our scope here; however, another reasonable clustering appears to be {1,2,3,4} and {5,6,7,8,9,10}. Burr (2006)

considers options for choosing the number of clusters for phylogenetic trees which describe the evolutionary history (branching order) of a group of taxa such as those being considered here.

The 12 SNP sites would typically be a very small fraction of the total number of sequenced sites. Suppose there are an additional 38 sites in the near physical vicinity on the genome but the SNP states are totally unrelated to the phenotype (disease present or absent) of interest. The middle plot of Figure 1 is based on all 50 SNP sites, and the resulting clustering is very ambiguous. The bottom plot is based on only SNP sites 1 and 6, which together would suggest the grouping {1,2,3,4} and {5,6,7,8,9,10}. The plotting symbols 1 to 10 are jittered slightly for readability.

Spurious Association

In many data mining applications, spurious associations are found, arising, for example, simply because so many tests for association are conducted, making it difficult to maintain a low false discovery rate. Spurious associations can also arise because effect A is linked to effect B only if factor C is at a certain level. For example, soft drink sales might link to sun poisoning incidence rates, not because of a direct causal link between soft drink sales and sun poisoning, but because both are linked to the heat index.

In GWAS, is it known that population subdivision can lead to spurious association between haplotype and phenotype (Liu et al., 2004). Bacanu et al. (2000) and Schaid (2004) show the potential power of using "genomic control" (GC) in a GWAS. The GC method adjusts for the nonindependence of subjects arising from the presence of subpopulations and other types of spurious associations. Schaid (2004) models the probability p that disease is present as a function of haplotype and covariates (possibly including subpopulation information), using, for example, a generalized linear model such as $y = \log(p/(1-p)) = f(X\beta) + \text{error}$, where haplotype and covariate information is captured in the X matrix, and β is estimated in order to relate X to y.

One common data mining activity is to evaluate many possible models relating predictors X to a scalar response y. Issues that arise include overfitting, error in X, nonrepresentative sampling of X (such as ignoring population substructure in a GWAS), feature selection, and selecting a functional form for $f()$. Standard data mining techniques such as cross validation or the use of held-out test data can help greatly in the general task of selecting an effective functional form for $f()$ that balances the tension between model complexity chosen to fit the training data well, and generalizability to new test data.

FUTURE TRENDS

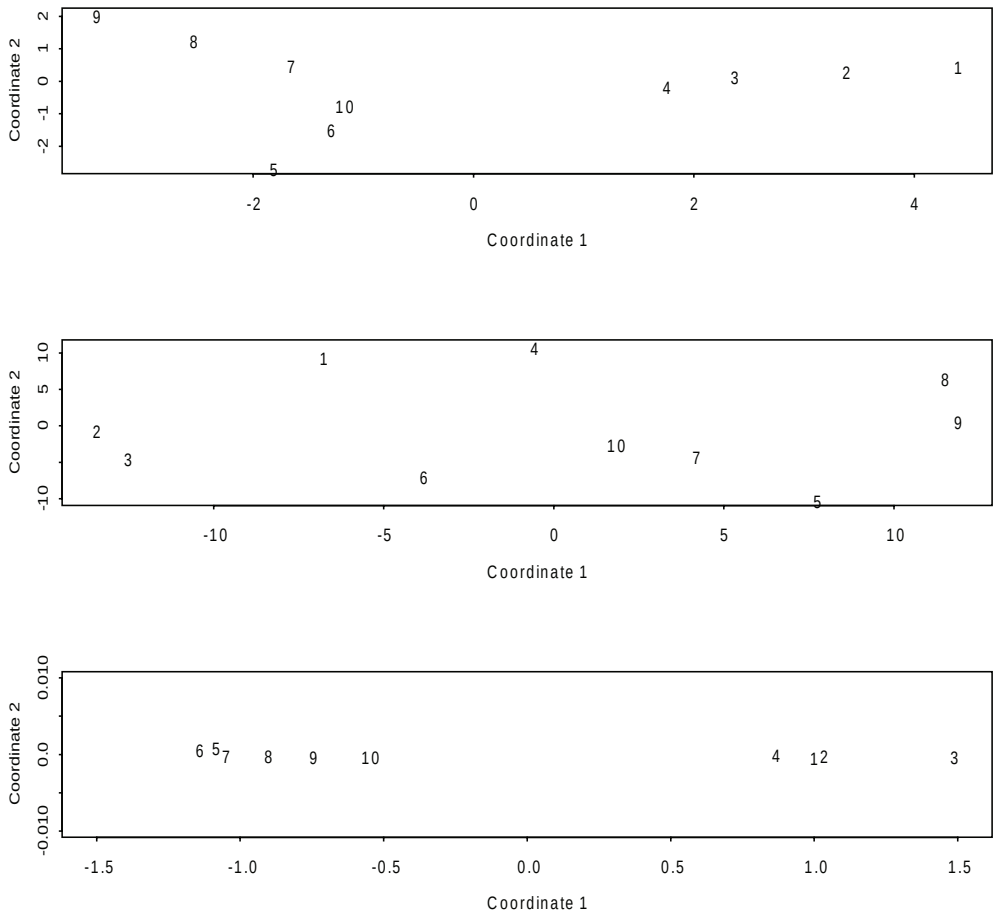
Searching for relations among multiple SNPs and phenotype will continue to use the techniques described. It is likely that evolutionary-model-based measures of haplotype similiarity will be increasingly invoked, and corresponding measures of genetic diversity in a sample will be developed.

Because it is now recognized that subpopulation structure can lead to spurious relations between phenotype (case or control, diseased or not, etc.) and allele frequency, genotype, and/or SNP state. Therefore, there is likely to be increasing attention to statistical sampling issues. For example, recent diabetes 2 literature has identified susceptibility loci on chromosomes 1q, 6, 11, and 14. However, Fingerlin et al. (2002) contradicts Hanis et al. (1996) regarding a susceptibility locus on chromosome 2. Such contradictions are not unusual, with many purported phenotype-genotype associations failing to be replicated. Resolving such a contradiction is related to the challenges previously described regarding finding effective haplotype "blocks" that can be replicated among samples. This particular contradiction appears to be explained by differences between Mexican-Americans and Finnish population samples.

Most common haplotypes occur in all human subpopulations; however, their frequencies differ among subpopulations. Therefore, data from several populations are needed to choose tag SNPs.

Pilot studies for the HapMap project found non-negligible differences in haplotype frequencies among population samples from Nigeria, Japan, China and the U.S. It is currently anticipated that GWAS using these four subpopulations should be useful for all subpopulations in the world. However, a parallel study is examining haplotypes in a set of chromosome regions in samples from several additional subpopulations. Several options to model subdivision and genetic exchange among partially subdivided populations are available and worth pursuing.

Figure 1. (top) Two-dimensional scaling plot of the matrix of pairwise Manhattan distances among the 10 example haplotypes. Haplotypes {1,2,3,4}, {5,6,10}, and {7,8,9} are candidate clusters. (middle) Same as top plot, except 0-1 values for each of an additional 38 SNP marker sites (a total of 50 SNP marker sites) are randomly added, rendering a very ambiguous clustering. (bottom) Same as top plot, except only marker sites 1 and 6 are used, resulting in a clear division into two clusters (the coordinate values are jittered slightly so the plotting symbols 1-10 can be read).



CONCLUSION

High throughput genomic technologies such as gene expression microarrays, SNP haplotype mapping as discussed here, array-based comparative genomic hybridization, etc., all involve genomic rather than single gene features of common human diseases.

Searching for relations among multiple SNPs and phenotype will continue to use the techniques described, including: genetics-model-based distance measures; feature selection; control of the false alarm rate; clustering in the context of phylogenetic tree building, and genomic control using genetic model fitting to protect against spurious association between halplotype and disease status.

Despite the challenges described here, GWAS successes have been reported, for example, for genotype-phenotype associations related to type 2 diabetes by Silander et al. (2005) and replicated by Zeggini et al. (2007).

REFERENCES

Bacanu S., Devlin B., Roeder K. (2000). The Power of Genomic Control. *American Journal of Human Genetics* 66, 1933-1944.

Burr T., Gattiker J., LaBerge G. (2002). Genetic Subtyping Using Cluster Analysis. *Special Interest Group on*

Knowledge Discovery and Data Mining Explorations 3, 33-42.

Burr T. (2006). Methods for Choosing Clusters in Phylogenetic Trees. *Encylopedia of Data Mining* John Wang, ed.

Fingerlin T., et al. (2002).Variation in Three Single Nucleotide Popymorphisms in the Callpain-10 Gene Not Associated with Type 2 Diabetes in a Large Finnish Cohort. *Diabetes* 51, 1644-1648.

Goldstein D., Cavalleri G.(2005). Understanding Human Diversity. *Nature* 437: 1241-1242.

Hanis C., et al. (1996). A Genome-Wide Search for Human Non-Insulin-Dependent (type 2) Diabetes Genes Reveals a Major Susceptibility Locus on Chromosome 2. *Nature Genetics* 13, 161-166.

HapMap: The International HapMap Consortium (2005). *Nature* 437, 1299-1320, www.hapmap.org.

Hannu T., Toivonen T., Onkamo P., Vasko K., Ollikainen V., Sevon P., Mannila H., Herr M., Kere J., (2000). Data Mining Applied to Linkage Disequilibrium Mapping. *American Journal of Human Genetics* 67, 133-145.

Liu N., Sawyer S., Mukherjee N., Pakstis A., Kidd J., Brookes Al, Zhao H. (2004). Haplotype Block Structures Show Significant Variation Among Populations. *Genetic Epidemiology* 27, 385-400.

Risch N., Merikangas K. (1996). The Future of Genetic Studies of Complex Human Diseases. *Science* 273, 1516-1517.

Schaid, D. (2004). Evaluating Associations of Haplotypes With Traits. *Genetic Epidemiology* 27, 348-364.

Silander K., et al. (2005). A Large Set of Finnish Affected Silbing Pair Families with Type 2 Diabetes Suggests Susceptibility Loci on Chromosomes 6, 11, and 14. *Diabetes* 53, 821-829.

Stram, D. (2004). Tag SNP Selection for Association Studies. *Genetic Epidemiology* 27, 365-374.

Sun Y., Jacobsen D., Kardia S. (2006). ChromoScan: a Scan Statistic Application for Identifying Chromosomal Regions in Genomic Studies. *Bioinformatics* 22(23), 2945-2947.

Tachmazidou I., Verzilli C., Delorio M. (2007). Genetic Association Mapping via Evolution-Based Clustering of Haplotypes. *PLoS Genetics* 3(7), 1163-1177.

Toivonen, H., Onkamo, P., Vasko, K., Ollikainen, V., Sevon, P., Mannila, Hl, Herr, M., and Kere, J. (2000). Data Mining Applied to Linkage Disequilibrium Mapping. *American Journal of Human Genetic*s 67, 133-1145.

Venables W., Ripley, D. (1999). Modern Applied Statistics with S-Plus, 3rd edition, Springer: New York.

Zeggini, E., et al. (2007). Replication of Genome-Wide Association Signals in UK Samples Reveals Risk Loci for Type II Diabetes. *Science* 316, 1336-1341.

Zhang K., Calabrese P., Nordborg M., and Sun F. (2002). Haplotype Block Structure and its Applications to Association Studies: Power and Study Designs. *American Journal of Human Genetics* 71, 1386-1394.

Zhang K., Jin L. (2003). HaploBlockFinder: Haplotype Block Analyses. *Bioinformatics* 19(10), 1300-1301.

KEY TERMS

Allele: A viable DNA (deoxyribonucleic acid) coding that occupies a given locus (site) on a chromosome.

Dynamic Programming: A method of solving problems that consist of overlapping subproblems by exploiting optimal substructure. Optimal substructure means that optimal solutions of subproblems can be used to find the optimal solution to the overall problem.

Haplotype: A haplotype is a set of single nucleotide polymorphisms (SNPs) on a single chromatid that are statistically associated.

Pearson χ^2 Test: A commonly-used statistical test used to detect association between two categorical variables, such as phenotype and allele type in our context.

Permutation Test: Is a statistical significance test in which a reference distribution is obtained by calculating many values of the test statistic by rearranging the labels (such as "case" and "control") on the observed data points.

Phenotype: The physical appearance of an organism, as opposed to its genotype**.** In our context, phenotype refers to whether the subject has the disease ("case") or does not ("control").

Phylogenetic Tree: A representation of the branching order and branch lengths of a collection of taxa, which, in its most common display form, looks like the branches of a tree.

Data Mining in Protein Identification by Tandem Mass Spectrometry

Haipeng Wang
Institute of Computing Technology & Graduate University of Chinese Academy of Sciences, China

INTRODUCTION

Protein identification (sequencing) by tandem mass spectrometry is a fundamental technique for proteomics which studies structures and functions of proteins in large scale and acts as a complement to genomics. Analysis and interpretation of vast amounts of spectral data generated in proteomics experiments present unprecedented challenges and opportunities for data mining in areas such as data preprocessing, peptide-spectrum matching, results validation, peptide fragmentation pattern discovery and modeling, and post-translational modification (PTM) analysis. This article introduces the basic concepts and terms of protein identification and briefly reviews the state-of-the-art relevant data mining applications. It also outlines challenges and future potential hot spots in this field.

BACKGROUND

Amino Acids, Peptides, and Proteins

An amino acid is composed of an amino group (NH_2), a carboxylic acid group (COOH), and a differentiating side chain (*R*). Each amino acid is represented by a letter from the English alphabet except B, J, O, U, X, and Z. A peptide or a protein is a chain that consists of amino acids linked together by peptide bonds. In this context, we refer to the products of enzymatic digestion of proteins as peptides.

Tandem Mass Spectrometry

Mass spectrometry (MS) is an analytical technique used to separate molecular ions and measure their mass-to-charge ratios (*m*/*z*). Tandem mass spectrometry (MS/MS), which can additionally fragment ionized molecules into pieces in a collision cell and measure the *m*/*z* values and ion current intensities of the pieces, is becoming increasingly indispensable for identifying complex protein mixtures in high-throughput proteomics.

MS-Based Protein Identification

According to the MS or MS/MS instrument adopted, there are two strategies for protein identification: peptide mass fingerprinting (PMF) and peptide fragment fingerprinting (PFF). The PFF approach is the focus of this article.

In a typical high-throughput PFF experiment, protein mixtures are digested with a site-specific protease (often trypsin) into complex peptide mixtures (e.g., the protein [AFCEFIVKLEDSE] digested into peptides [AFCEFIVK] and [LEDSE]). The resulted peptides are separated typically by liquid chromatography (LC) and sequentially fed into a MS/MS instrument. In this instrument the separated peptides are ionized with one or more units of charges, selected according to their *m*/*z* values, and broken into pieces by low-energy collision-induced dissociation (CID) resulting in various types of fragment ions (Figure 1). The fragment ions are measured to obtain a bundle of spectral peaks each comprising an *m*/*z* and an intensity values. Peaks plus the *m*/*z* value and charge state of a peptide ion constitutes a peptide MS/MS spectrum (Figure 2). It is not necessary that every possible fragment ion appear in the spectrum.

Figure 1. The nomenclature for various types of fragment ions

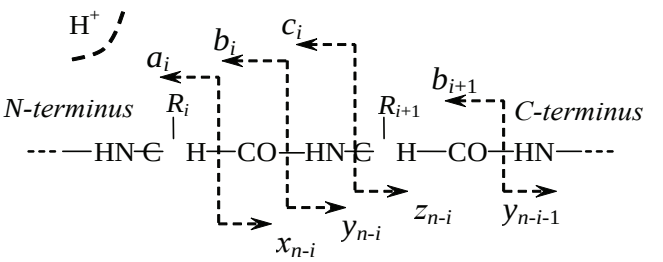
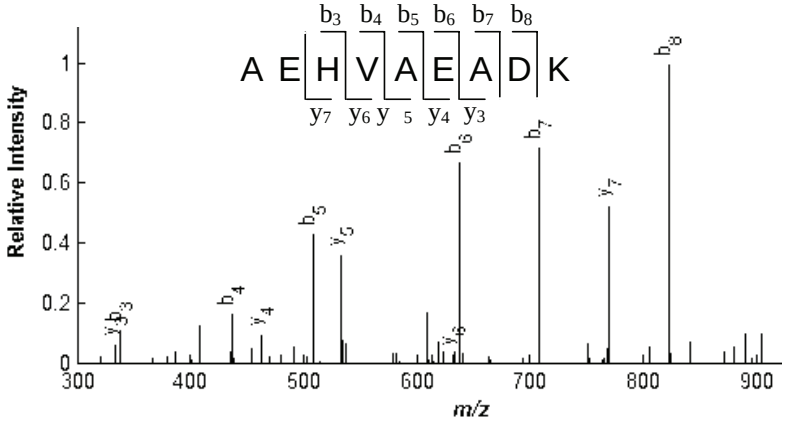


Figure 2. An example of a MS/MS spectrum and its annotation



Peptide identification is the core of protein sequencing in that a protein sequence is identified from its constituent peptides with the help of protein databases. There are three different approaches for PFF peptide identification: database searching, *de novo* sequencing, and sequence tagging.

Database searching is the most widely used method in high-throughput proteomics. It correlates an experimental MS/MS spectrum with theoretical ones generated from a list of peptides digested *in silicon* from proteins in a database. As annotated MS/MS spectra increase, a spectrum library searching method which compares an experimental spectrum to previously identified ones in the reference library has recently proposed (Craig et al., 2006; Frewen et al., 2006). *De novo* sequencing tries to infer the complete peptide sequence directly from the *m*/*z* differences between peaks in a MS/MS spectrum without any help of databases. Sequence tagging yields one or more partial sequences by *de novo* sequencing, then finds homologous sequences in the protein database and finally scores the homologous candidate sequences to obtain the true peptide.

Robust data preprocessing, scoring scheme, validation algorithm, and fragmentation model are the keys to all of the above three approaches.

MAIN FOCUS

Preprocessing of MS/MS Spectra

The purpose of data preprocessing is to retain the potentially contributing data while removing the noisy or misleading ones as many as possible to improve the reliability and efficiency of protein identification. In general, only a relatively small fraction (say, <30%) of a large MS/MS spectra set is identified confidently due to some reasons. First, there exist some chemical or electronical noises in a spectrum resulting in random matches to peptides. Second, the peptide ion charge state of a spectrum is usually ambiguous due to the inability of instruments. Third, there are considerable duplicate spectra most likely representing a unique peptide. Four, many poor-quality spectra often have no results which can be filtered in advance using classification or regression methods. Clustering algorithms (Tabb et al., 2003; Beer et al., 2004), linear discriminant analysis (LDA) (Nesvizhskii et al., 2006), quadratic discriminant function (Xu et al., 2005), SVM (Bern et al., 2004; Klammer et al., 2005), linear regression (Bern et al., 2004), Bayesian rules (Colinge et al., 2003a; Flikka et al., 2006), decision tree and random forest (Salmi et al., 2006) have been widely used to address the above problems.

Scoring Scheme for Peptide Candidates

The goal of scoring is to find the true peptide-spectrum match (PSM). In principle there are two implementation frameworks for this goal: descriptive or probabilistic.

In descriptive frameworks, an experimental and a theoretical MS/MS spectra are represented as vectors $\mathbf{S} = (s_1,s_2,...,s_n)$ and $\mathbf{T} = (t_1,t_2,...,t_n)$, respectively, where n denotes the number of predicted fragments, s_i and t_i are binary values or the observed and predicted intensity

values of the *i*th fragment, respectively. The similarity between experimental and theoretical spectrum can be measured using shared peak count, spectral dot product (SDP) (Field et al., 2002) or cross-correlation (Eng et al., 1994). Fu et al. (2004) extends the SDP method by using kernel tricks to incorporate the information of consecutive fragment ions. SEQUEST and pFind are two representative software programs in this category.

In probabilistic frameworks, one strategy is to calculate a probability $P(\mathbf{S} \mid p, \ random)$ under the null hypothesis that the match between a peptide p and an experimental spectrum **S** is a random event (Perkins et al., 1999; Sadygov & Yates, 2003; Geer et al., 2004; Narasimhan et al., 2005). The hypergeometric distribution (Sadygov & Yates, 2003), Poisson distribution (Geer et al., 2004), and multinomial distribution (Narasimhan et al., 2005) was used to model the random matches. In another strategy, a probability $P(\mathbf{S} \mid p, correct)$ under the alternative hypothesis that a peptide p correctly generates a spectrum **S** is derived using certain fragmentation models (Bafna & Edwards, 2001; Zhang et al., 2002; Havilio et al., 2003; Wan et al., 2006). Wan et al. (2006) combined information on mass accuracy, peak intensity, and correlation among ions into a hidden Markov model (HMM) to calculate the probability of a PSM. As a combination of the above two ideas, a likelihood-ratio score can be obtained for assessing the relative tendency of peptide identifications being true or random (Colinge et al., 2003b; Elias et al., 2004). Elias et al. (2004) used probabilistic decision trees to model the random and true peptide matches respectively based on many features constructed from peptides and spectra and achieved significant improvement in suppressing false positive matches. Mascot and Phynex are two representative software programs in this framework (Perkins et al., 1999; Colinge et al., 2003b).

Validation of Peptide Identifications

Any of the scoring schemes sorts candidate peptides according to the score reflecting the similarity between the spectrum and the peptide sequence. Even if a peptide score is ranked at the first place, the score is not always sufficient for evaluating the correctness of a PSM. As a result, peptide identifications need to be carefully scrutinized manually (Chen et al., 2005) or automatically.

The common validation method is to use a threshold to filter peptide identifications (Elias et al., 2005). Its straightforward and operable features have facilitated proteomics experiments but it can not well balance the tradeoff between sensitivity and precision of peptide identification.

Some methods report the significance of peptide identifications by the E-value proposed in Fenyo & Beavis (2003). Keller et al. (2002) first used an LDA to separate true matches from random ones based on SEQUEST outputs and then derive a probability by applying an expectation-maximization (EM) algorithm to fit the Gaussian and gamma distributions of the LDA scores of true and random matches respectively.

Actually the LDA method in Keller et al. (2002) is just one kind of validation algorithms based on pattern classification methods such as SVM (Anderson et al., 2003), neural network (Baczek et al., 2004), decision tree and random forest (Ulintz et al., 2006), in which features were all extracted from SEQUEST outputs. Wang et al. (2006) developed an SVM scorer combining together the scoring and the validation processes of peptide identification. It can not only serve as a post-processing module for any peptide identification algorithm but also be an independent system. Robust feature extraction and selection form the basis of this method.

Peptide Fragmentation Models Under CID

The study of the mechanisms underlying peptide fragmentation will lead to more accurate peptide identification algorithms. Its history can be traced back to 1980's when the physicochemical methods were popular (Zhang, 2004; Paizs & Suhuai 2005). Recently, statistical methods have been applied to fragmentation pattern discovery (Wysocki et al. 2000; Kapp et al. 2003). The mobile-proton model, currently a most comprehensive model, was formed, improved, justified, and reviewed in the above references. Kapp et al. (2003) and Zhang (2004) are two representative methods based on two respective different strategies: top-down statistical and bottom-up chemical methods. The former explored the fragmentation trends between any two amino acids from a large database of identified PSMs and then applied a linear regression to fit the spectral peak intensities. The latter made an assumption that a certain spectrum is the result of the

competition of many different fragmentation pathways for a peptide under CID and a kinetic model based on the knowledge from the physicochemical domain was learnt from a training set of PSMs.

As Paizs & Suhai (2005) stated, the chemical and statistical methods will respectively provide a necessary framework and a quantitative description for the simulation of peptide fragmentation. Some databases of validated PSMs have been established to provide training data sets for discovery and modeling of peptide fragmentation mechanisms (Martens et al., 2005; McLaughlin et al., 2006).

FUTURE TRENDS

An ensemble scorer that combines together different types of scoring scheme will bring more confident identifications of MS/MS spectra. A peptide identification result with the corresponding estimated false positive rate is imperative. To assign peptides to proteins, there is an increased need for reliable protein inference algorithms with probability scores (Feng et al., 2007).

The algorithms combining database searching and *de novo* sequencing will be prevailing for reducing time complexity and improving robustness of current protein identification algorithms (Bern et al., 2007).

Peptide fragmentation modeling is still an open problem for data mining, proteomics and mass spectrometry communities. Nowadays, there is relatively little work involving this study. With more and more exposure to the mechanisms of peptide fragmentation from the physicochemical domain and intensively increased identified MS/MS spectra, the fragmentation model will be disclosed in a quantitative way.

Effective, efficient, practicable, especially unrestrictive PTM identification algorithms (Tanner et al., 2006; Havilio & Wool, 2007) are coming into focus in that most proteins undergo modification after being synthesized and PTM plays an extremely important role in protein function.

CONCLUSION

Protein identification based on MS/MS spectra is an interdisciplinary subject challenging both biologists and computer scientists. More effective data preprocessing, sensitive PSM scoring, and robust results validation are still the central tasks of protein identification, while the fields of peptide fragmentation modeling and PTM identification are making new opportunities for the development and application of data mining methods.

REFERENCES

Anderson, D. C., Li, W., Payan D. G., Noble, W. S. (2003). A new algorithm for the evaluation of shotgun peptide sequencing in proteomics: Support vector machine classification of peptide MS/MS spectra and SEQUEST scores. *Journal of Proteome Research*, 2(2), 137-146.

Baczek, T., Bucinski, A., Ivanov, A. R., & Kaliszan, R. (2004). Artificial neural network analysis for evaluation of peptide MS/MS spectra in proteomics. *Analytical Chemistry*, 76(6), 1726-1732.

Bafna, V., & Edwards, N. (2001). SCOPE: a probabilistic model for scoring tandem mass spectra against a peptide database. *Bioinformatics*, 17(Suppl. 1), S13-S21.

Beer, I., Barnea, E., Ziv, T., & Admon, A. (2004). Improving large-scale proteomics by clustering of mass spectrometry data. *Proteomics*, 4(4), 950-960.

Bern, M., Goldberg, D., McDonald, W. H., & Yates, J. R. III. (2004). Automatic quality assessment of peptide tandem mass spectra. *Bioinformatics*, 20(Suppl. 1), I49-I54.

Bern, M., Cai, Y., & Goldberg, D. (2007). Lookup peaks: a hybrid of de novo sequencing and database search for protein identification by tandem mass spectrometry. *Analytical Chemistry*, 79(4), 1393-1400.

Chen, Y., Kwon, S. W., Kim, S. C., & Zhao, Y. (2005). Integrated approach for manual evaluation of peptides identified by searching protein sequence databases with tandem mass spectra. *Journal of Proteome Research*, 4(3), 998-1005.

Colinge, J., Magnin, J., Dessingy, T., Giron, M., & Masselot, A. (2003a). Improved peptide charge state assignment. *Proteomics*, 3(8), 1434-1440.

Colinge, J., Masselot, A., Giron, M., Dessingy, T., & Magnin, J. (2003b). OLAV: towards high-throughput tandem mass spectrometry data identification. *Proteomics*, 3(8), 1454-1463.

Craig, R., Cortens, J. C., Fenyo, D., & Beavis, R. C. (2006). Using annotated peptide mass spectrum libraries for protein identification. *Journal of Proteome Research*, 5(8), 1843-1849.

Elias, J. E., Gibbons, F. D., King, O. D., Roth, F. P., & Gygi, S. P. (2004). Intensity-based protein identification by machine learning from a library of tandem mass spectra. *Nature Biotechnology*, 22(2), 214-219.

Elias, J. E., Haas, W., Faherty, B. K., & Gygi, S. P. (2005). Comparative evaluation of mass spectrometry platforms used in large-scale proteomics investigations. *Nature Methods*, 2(9), 667-675.

Eng, J. K., McCormack, A. L., & Yates, J. R. III. (1994). An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. *Journal of the American Society for Mass Spectrometry*, 5(11), 976-989.

Feng, J., Naiman, D.Q., & Cooper, B. (2007). Probability model for assessing proteins assembled from peptide sequences inferred from tandem mass spectrometry data. *Analytical Chemistry*, 79(10), 3901-3911.

Fenyo, D., & Beavis, R. C. (2003). A method for assessing the statistical significance of mass spectrometry-based protein identifications using general scoring schemes. *Analytical Chemistry*, 75(4), 768-774.

Field, H. I., Fenyö, D., & Beavis, R. C. (2002). RADARS, a bioinformatics solution that automates proteome mass spectral analysis, optimises protein identification, and archives data in a relational database. *Proteomics*, 2(1), 36-47.

Flikka, K., Martens, L., Vandekerckhove, J., Gevaert, K. & Eidhammer, I. (2006). Improving the reliability and throughput of mass spectrometry-based proteomics by spectrum quality filtering. *Proteomics*, 6(7), 2086-2094.

Frewen, B. E., Merrihew, G. E., Wu, C. C., Noble, W. S., & MacCoss, M. J. (2006). Analysis of peptide MS/MS spectra from large-scale proteomics experiments using spectrum libraries. *Analytical Chemistry*, 78, 5678-5684.

Fu, Y., Yang, Q., Sun, R., Li, D., Zeng, R., Ling, C. X., & Gao, W. (2004). Exploiting the kernel trick to correlate fragment ions for peptide identification via tandem mass spectrometry. *Bioinformatics*, 20(12), 1948-1954.

Geer, L. Y., Markey, S. P., Kowalak, J. A., Wagner, L., Xu, M., Maynard, D. M., Yang, X., Shi, W., & Bryant, S. H. (2004). Open mass spectrometry search algorithm. *Journal of Proteome Research*, 3(5), 958-964.

Havilio, M., Haddad, Y., & Smilansky, Z. (2003). Intensity-based statistical scorer for tandem mass spectrometry. *Analytical Chemistry*, 75(3), 435-444.

Havilio, M., & Wool, A. (2007). Large-scale unrestricted identification of post-translation modifications using tandem mass spectrometry. Analytical Chemistry, 79(4), 1362-1368.

Kapp, E. A., Schutz, F., Reid, G. E., Eddes, J. S., Moritz, R. L., O'Hair, R. A., Speed, T.P., & Simpson, R. J. (2003). Mining a tandem mass spectrometry database to determine the trends and global factors influencing peptide fragmentation. *Analytical Chemistry*, 75(22), 6251-6264.

Keller, A., Nesvizhskii, A., Kolker, E., & Aebersold, R. (2002). Empirical statistical model to estimate the accuracy of peptide identifications made by MS/MS and database search. *Analytical Chemistry*, 74(20), 5383-5392.

Klammer, A. A., Wu, C. C., MacCoss, M. J., & Noble, W. S. (2005). Peptide charge state determination for low-resolution tandem mass spectra. In P. Markstein & Y. Xu (Eds.), *Proceedings of IEEE Computational Systems Bioinformatics Conference 2005* (pp. 175-185). London, UK: Imperial College Press.

Martens, L., Hermjakob, H., Jones, P., Adamski, M., Taylor, C., States, D., Gevaert, K., Vandekerckhove, J., & Apweiler, R. (2005). PRIDE: the proteomics identifications database. *Proteomics*, 5(13), 3537-3545.

McLaughlin, T., Siepen, J. A., Selley, J., Lynch, J. A., Lau, K. W., Yin, H., Gaskell, S. J., & Hubbard, S. J. (2006). PepSeeker: a database of proteome peptide identifications for investigating fragmentation patterns. *Nucleic Acids Research*, 34(database issue), D649-D654.

Narasimhan, C., Tabb, D. L., Verberkmoes, N. C., Thompson, M. R., Hettich, R. L., & Uberbacher, E. C. (2005). MASPIC: intensity-based tandem mass spectrometry scoring scheme that improves peptide identification at high confidence. *Analytical Chemistry*, 77(23), 7581-7593.

Nesvizhskii, A. I., Roos, F. F., Grossmann, J., Vogelzang, M., Eddes, J. S., Gruissem, W., Baginsky, S., & Aebersold, R. (2006). Dynamic Spectrum Quality Assessment and Iterative Computational Analysis of Shotgun Proteomic Data. *Molecular & Cellular Proteomics*, 5(4), 652-670.

Paizs, B., & Suhai, S. (2005). Fragmentation pathways of protonated peptides. *Mass Spectrometry Reviews*, 24(4), 508-548.

Perkins, D. N., Pappin, D. J., Creasy, D. M., & Cottrell, J. S. (1999). Probability-based protein identification by searching sequence databases using mass spectrometry data. *Electrophoresis*, 20(18), 3551-3567.

Sadygov, R. G., & Yates, J. R. III. (2003). A hypergeometric probability model for protein identification and validation using tandem mass spectral data and protein sequence databases. *Analytical Chemistry*, 75(15), 3792-3798.

Salmi, J., Moulder, R., Filen, J. J, Nevalainen, O. S., Nyman, T. A., Lahesmaa, R., & Aittokallio, T. (2006). Quality classification of tandem mass spectrometry data. *Bioinformatics*, 22(4), 400-406.

Tabb, D. L., MacCoss, M. J., Wu, C. C., Anderson, S. D., & Yates, J. R. III. (2003). Similarity among tandem mass spectra from proteomic experiments: detection, significance, and utility. *Analytical Chemistry*, 75(10), 2470-2477.

Tanner, S., Pevzner, P. A., & Bafna, V. (2006). Unrestrictive identification of post-translational modifications through peptide mass spectrometry. Nature Protocols, 1(1), 67-72.

Ulintz, P. J., Zhu, J., Qin, Z. S., & Andrews, P. C. (2006). Improved classification of mass spectrometry database search results using newer machine learning approaches. *Molecular & Cellular Proteomics*, 5(3), 497-509.

Wan, Y., Yang, A., & Chen, T. (2006). PepHMM: a hidden Markov model based scoring function for mass spectrometry database search. *Analytical Chemistry*, 78(2), 432-437.

Wang, H., Fu, Y., Sun, R., He, S., Zeng, R., & Gao, W. (2006). An SVM scorer for more sensitive and reliable peptide identification via tandem mass spectrometry. In R. B. Altman, T. Murray, T. E. Klein, A. K. Dunker, & L. Hunter (Eds.), *Pacific Symposium on Biocomputing 11* (pp. 303-314). Singapore: World Scientific Publishing Co. Pte. Ltd.

Wysocki, V. H., Tsaprailis, G., Smith, L. L., & Breci, L. A. (2000). Mobile and localized protons: A framework for understanding peptide dissociation. *Journal of Mass Spectrometry*, 35(12), 1399-1406.

Xu, M., Geer, L. Y., Bryant, S. H., Roth, J. S., Kowalak, J. A., Maynard, D. M., & Markey, S. P. (2005). Assessing data quality of peptide mass spectra obtained by quadrupole ion trap mass spectrometry. *Journal of Proteome Research*, 4(2), 300-305.

Zhang, N., Aebersold, R., & Schwikowski, B. (2002). ProbID: A probabilistic algorithm to identify peptides through sequence database searching using tandem mass spectral data. *Proteomics*, 2(10), 1406–1412.

Zhang, Z. (2004). Prediction of low-energy collision-induced dissociation spectra of peptides. *Analytical Chemistry*, 76(14), 3908-3922.

KEY TERMS

Bioinformatics: An interdisciplinary field that studies how to apply informatics to explore biological data.

Classification: A process that divides samples into several homogeneous subsets according to a model learnt from a training set of samples with known class labels.

Clustering: A process that groups samples into several subsets based on similarity (often distance measure).

Feature Extraction: A process of constructing features for describing a sample as characteristic as possible for clustering, classification or regression.

Feature Selection: A process of choosing an optimal subset of features from the existing features according to a specific purpose.

Precision: A ratio of the number of true positive samples to the total number of both false positive and true positive samples in a binary classification task.

Proteomics: A complement to genomics to study all the levels of proteins in a cell or a tissue under various physiological conditions in a high-throughput way.

Regression: A statistical method to determine the relationship between two variables.

Sensitivity (Recall): A ratio of the number of true positive samples to the total number of both false negative and true positive samples in a binary classification task.

Data Mining in Security Applications

Aleksandar Lazarevic
United Technologies Research Center, USA

INTRODUCTION

In recent years, research in many security areas has gained a lot of interest among scientists in academia, industry, military and governmental organizations. Researchers have been investigating many advanced technologies to effectively solve acute security problems. Data mining has certainly been one of the most explored technologies successfully applied in many security applications ranging from computer and physical security and intrusion detection to cyber terrorism and homeland security. For example, in the context of homeland security, data mining can be a potential means to identify terrorist activities, such as money transfers and communications, and to identify and track individual terrorists themselves, such as through travel and immigration records (Seifert, 2007). In another data mining's success story related to security, credit card fraud detection, all major credit card companies mine their transaction databases, looking for spending patterns that indicate a stolen card. In addition, data mining has also effectively been utilized in many physical security systems (e.g. in efficient system design tools, sensor fusion for false alarm reduction) and video surveillance applications, where many data mining based algorithms have been proposed to detect motion or intruder at monitored sites or to detect suspicious trajectories at public places.

This chapter provides an overview of current status of data mining based research in several security applications including cyber security and intrusion detection, physical security and video surveillance.

BACKGROUND

As the cost of the information processing and Internet accessibility falls, more and more organizations are becoming vulnerable to a wide variety of cyber threats. It has become increasingly important recently to make our information systems, especially those used for critical functions in the military and commercial sectors, resistant to and tolerant of such attacks. The conventional security mechanisms, such as firewalls, authentication mechanisms, Virtual Private Networks (VPN) almost always have inevitable vulnerabilities and they are usually insufficient to ensure complete security of the infrastructure and to ward off attacks that are continually being adapted to exploit the system's weaknesses. This has created the need for security technology, called intrusion detection that includes identifying malicious actions that compromise the integrity, confidentiality, and availability of information resources.

MAIN THRUST OF THE CHAPTER

Cyber Security and Intrusion Detection

Traditional intrusion detection systems (IDSs) are based on extensive knowledge of signatures (rule descriptions) of known attacks. However, the signature database has to be manually revised for each new type of intrusion that is discovered. In addition, signature-based methods cannot detect emerging cyber threats, since by their very nature these threats are launched using previously unknown attacks. These limitations have led to an increasing interest in intrusion detection techniques based upon data mining. Data mining techniques for cyber security and intrusion detection generally fall into one of two categories: misuse detection, and anomaly detection.

Misuse Detection

In misuse detection techniques, each instance in a data set is labeled as 'normal' or 'attack/intrusion' and a learning algorithm is trained over the labeled data. Unlike signature-based IDSs, data mining based misuse detection models are created automatically, and can be more sophisticated and precise than manually created signatures. In spite of the fact that misuse detection models have high degree of accuracy in detecting known attacks and their variations, their obvious drawback is

the inability to detect attacks whose instances have not yet been observed. In addition, labeling data instances as normal or intrusive may require enormous time for many human experts.

Since standard data mining techniques are not directly applicable to the problem of intrusion detection due to skewed class distribution (attacks/intrusions correspond to a much smaller, i.e. rarer class, than the class representing normal behavior) and streaming nature of data (attacks/intrusions very often represent sequence of events), a number of researchers have developed specially designed data mining algorithms suitable for intrusion detection. Research in misuse detection has focused mainly on classifying network intrusions using various standard data mining algorithms (Barbara, 2001; Lee, 2001), rare class predictive models (Joshi, 2001) and association rules (Barbara, 2001; Lee, 2000; Manganaris, 2000).

MADAM ID (Lee, 2000; Lee, 2001) was one of the first projects that applied data mining techniques to the intrusion detection problem. In addition to the standard features that were available directly from the network traffic (e.g. duration, start time, service), three groups of constructed features (content-based features that describe intrinsic characteristics of a network connection (e.g. number of packets, acknowledgments, data bytes from source to destination), time-based traffic features that compute the number of connections in some recent time interval and connection based features that compute the number of connections from a specific source to a specific destination in the last *N* connections) were also used by the RIPPER algorithm to learn intrusion detection rules. Other classification algorithms that are applied to the intrusion detection problem include standard decision trees (Bloedorn, 2001), modified nearest neighbor algorithms (Ye, 2001b), fuzzy association rules (Bridges, 2000), neural networks (Dao, 2002; Kumar, 2007; Zhang, 2005), support vector machines (Chen, 2005; Kim, 2005), naïve Bayes classifiers (Bosin, 2005; Schultz, 2001), genetic algorithms (Bridges, 2000; Kim, 2005; Li, 2004), genetic programming (Mukkamala, 2003), etc. Most of these approaches attempt to directly apply specified standard techniques to publicly available intrusion detection data sets (Lippmann, 1999; Lippmann, 2000), assuming that the labels for normal and intrusive behavior are already known. Since this is not realistic assumption, misuse detection based on data mining has not been very successful in practice.

Anomaly Detection

Anomaly detection creates profiles of normal "legitimate" computer activity (e.g. normal behavior of users (regular e-mail reading, web browsing, using specific software), hosts, or network connections) using different techniques and then uses a variety of measures to detect deviations from defined normal behavior as potential anomaly. Anomaly detection models often learn from a set of "normal" (attack-free) data, but this also requires cleaning data from attacks and labeling only normal data records. Nevertheless, other anomaly detection techniques detect anomalous behavior without using any knowledge about the training data. Such models typically assume that the data records that do not belong to the majority behavior correspond to anomalies.

The major benefit of anomaly detection algorithms is their ability to potentially recognize unforeseen and emerging cyber attacks. However, their major limitation is potentially high false alarm rate, since detected deviations may not necessarily represent actual attacks, but new or unusual, but still legitimate, network behavior.

Anomaly detection algorithms can be classified into several groups: (i) statistical methods; (ii) rule based methods; (iii) distance based methods (iv) profiling methods and (v) model based approaches (Lazarevic, 2005b). Although anomaly detection algorithms are quite diverse in nature, and thus may fit into more than one proposed category, most of them employ certain artificial intelligence techniques.

Statistical methods. Statistical methods monitor the user or system behavior by measuring certain variables over time (e.g. login and logout time of each session). The basic models keep averages of these variables and detect whether thresholds are exceeded based on the standard deviation of the variable. More advanced statistical models compute profiles of long-term and short-term user activities by employing different techniques, such as Chi-square (χ^2) statistics (Ye, 2001a), probabilistic modeling (Yamanishi, 2000), Markov-chain models (Zanero, 2006) and likelihood of data distributions (Eskin, 2000).

Distance based methods. Most statistical approaches have limitation when detecting outliers in higher dimensional spaces, since it becomes increasingly difficult and inaccurate to estimate the multidimensional distributions of the data points. Distance based approaches attempt to overcome these limitations by

detecting outliers using computed distances among points. Several distance based outlier detection algorithms (Lazarevic, 2003; Wang, 2004) as well as ensemble of outlier detection methods (Lazarevic, 2005a) that have been recently proposed for detecting anomalies in network traffic are based on computing the full dimensional distances of points from one another using all the available features, and on computing the densities of local neighborhoods. MINDS (Minnesota Intrusion Detection System) (Chandola 2006) employs outlier detection algorithms to routinely detect various intrusions that could not be detected using widely used traditional IDSs. These detected intrusions include various suspicious behavior and policy violations (e.g. identifying compromised machines, covert channels, illegal software), variations of worms (e.g., variations of slapper and slammer worms), as well as scanning activities (e.g., scans for unknown vulnerable ports, distributed scans). Many of these detected intrusions (scanning for Microsoft DS service on port 445/TCP) could not be detected using widely used traditional IDSs.

Several clustering based techniques, such as fixed-width and canopy clustering (Eskin, 2002), as well as density-based and grid-based clustering (Leung, 2005), have also been used to detect network intrusions as small clusters comparing to the large ones that corresponded to the normal behavior. In another interesting approach (Fan, 2001), artificial anomalies in the network intrusion detection data are generated around the edges of the sparsely populated "normal" data regions, thus forcing the learning algorithm to discover the specific boundaries that distinguish these "anomalous" regions from the "normal" data.

Rule based systems. Rule based systems were used in earlier anomaly detection based IDSs to characterize normal behavior of users, networks and/or computer systems by a set of rules. Examples of such rule based IDSs include ComputerWatch (Dowell, 1990) and Wisdom & Sense (Liepins, 1992). Recently, there were also attempts to automatically characterize normal behavior using association rules (Lee, 2001).

Profiling methods. In profiling methods, profiles of normal behavior are built for different types of network traffic, users, programs etc., and deviations from them are considered as intrusions. Profiling methods vary greatly ranging from different data mining techniques to various heuristic-based approaches.

For example, ADAM (Audit Data and Mining) (Barbara, 2001) is a hybrid anomaly detector trained on both attack-free traffic and traffic with labeled attacks. The system uses a combination of association rule mining and classification to discover novel attacks in tcpdump data by using the pseudo-Bayes estimator. Recently reported PHAD (packet header anomaly detection) (Mahoney, 2002) monitors network packet headers and builds profiles for 33 different fields (e.g., IP protocol, IP src/dest, IP/TCP src/dest ports) from these headers by observing attack free traffic and building contiguous clusters for the values observed for each field. ALAD (application layer anomaly detection) (Mahoney, 2002) uses the same method for calculating the anomaly scores as PHAD, but it monitors TCP data and builds TCP streams when the destination port is smaller than 1024.

Finally, there have also been several recently proposed commercial products that use profiling based anomaly detection techniques, such as Mazu Network Behavior Analysis system (Mazu Networks, 2007) and Peakflow X (Arbor Networks, 2007).

Model based approaches. Many researchers have used different types of data mining models such as replicator neural networks (Hawkins, 2002) or unsupervised support vector machines (Eskin, 2002; Lazarevic, 2003) to characterize the normal behavior of the monitored system. In the model-based approaches, anomalies are detected as deviations from the model that represents the normal behavior. Replicator four-layer feed-forward neural network reconstructs input variables at the output layer during the training phase, and then uses the reconstruction error of individual data points as a measure of outlyingness, while unsupervised support vector machines attempt to find a small region where most of the data lies, label these data points as a normal behavior, and then detect deviations from learned models as potential intrusions.

Physical Security and Video Surveillance

New types of intruder and terrorist threats around the world are accelerating the need for new, integrated security technologies that combine video and security analytics to help responders act faster during emergencies and potentially preempt attacks. This integrated vision of security is transforming the industry, revitalizing traditional surveillance markets and creating new opportunities for IT professionals. Networked video

surveillance systems apply powerful analytics to the video data and are known as intelligent video. Integration of many data sources into a common analytical frame of reference provides the grist for sophisticated forensic data mining techniques and unprecedented real-time awareness that can be used to quickly see the whole picture of what is occurring and trigger appropriate action.

There has been a surge of activity on using data mining techniques in video surveillance, based on extension of classification algorithms, similarity search, clustering and outlier detection techniques. Classification techniques for video surveillance include support vector machines to classify objects in far-field video images (Bose, 2002), Bhattacharya coefficients for automatic identification of events of interest, especially of abandoned and stolen objects in a guarded indoor environment (Ferrando 2006), and Mahalanobis classifiers for detecting anomalous trajectories as time series modeled using orthogonal basis function representations (Naftel 2006). In addition, non-metric similarity functions based on the Longest Common Subsequence (LCSS) have been used to provide an intuitive notion of similarity between video object trajectories in two or three dimensional space by giving more weight to the similar portions of the sequences (Vlachos 2002). The new measure has been demonstrated to be superior comparing to widely used Euclidean and Time Warping distance functions.

In clustering-based video surveillance techniques, trajectory clustering has been often active area of investigation. In (Foresti, 2005), a trajectory clustering method suited for video surveillance and monitoring systems was described, where the clusters are dynamic and built in real-time as the trajectory data is acquired. The obtained clusters have been successfully used both to give proper feedback to the low-level tracking system and to collect valuable information for the high-level event analysis modules. Yanagisawa and Satoh (Yanagisawa, 2006) have proposed using clustering to perform similarity queries between the shapes of moving object trajectories. Their technique is an extension of similarity queries for time series data, combined with velocities and shapes of objects.

Finally, detecting outliers from video data has gained a lot of interest recently. Researchers have been investigating various techniques including point distribution models and Hotteling's T2 statistics to detect outliers in trajectory data (De Meneses, 2005), Ullman and Basri's linear combination (LC) representation for outlier detection in motion tracking with an affine camera (Guo, 2005), and principal component analysis of 3D trajectories combined with incremental outlier detection algorithms to detect anomalies in simulated and real life experimental videos (Pokrajac, 2007).

FUTURE TRENDS

Along the changes around the world, the weapons and tools of terrorists and intruders are changing as well, thus leaving an opportunity for many artificial intelligence technologies to prove their capabilities in detecting and deterring criminals. Supported by FBI in the aftermath of September 11, data mining will start to be immensely involved in many homeland security initiatives. In addition, due to recent trend of integrating cyber and physical security, data from different monitoring systems will be integrated as well thus making a fruitful ground for various data mining techniques. There are an increasing number of government organizations and companies interested in using data mining techniques to detect physical intrusions, identify computer and network attacks, and provide video content analysis.

CONCLUSION

Data mining techniques for security application have improved dramatically over time, especially in the past few years, and they increasingly become an indispensable and integral component of any comprehensive enterprise security program, since they successfully complement traditional security mechanisms. Although a variety of techniques have been developed for many applications in security domain, there are still a number of open research issues concerning the vulnerability assessment, effectiveness of security systems, false alarm reduction and prediction performance that need to be addressed.

REFERENCES

Adderley, R. (2003). Data Mining in the West Midlands Police: A Study of Bogus Official Burglaries. "Investigative Data Mining for Security and Criminal Detection", Jesus Mena, 24-37.

Arbor Networks. (2007). Peakflow X, http://www.arbornetworks.com/products_x.php.

Barbara, D., Wu, N. & Jajodia, S. (2001). Detecting Novel Network Intrusions Using Bayes Estimators. In Proceedings of the *First SIAM Conference on Data Mining*, Chicago, IL.

Bloedorn, E., et al. (2001). Data Mining for Network Intrusion Detection: How to Get Started. www.mitre.org/work/tech_papers/tech_papers_01/bloedorn_datamining, *MITRE Technical Report*.

Bose, B. (2002). Classifying Tracked Objects in Far-Field Video Surveillance, Masters' Thesis, MIT, Boston, MA.

Bosin, A., Dessi, N., Pes, B. (2005). Intelligent Bayesian Classifiers in Network Intrusion Detection, In Proceedings of the *18th international conference on Innovations in Applied Artificial Intelligence*, 445-447, Bari, Italy.

Bridges, S. & Vaughn, R. (2000). Fuzzy Data Mining and Genetic Algorithms Applied to Intrusion Detection, In Proceedings of the *23rd National Information Systems Security Conference*, Baltimore, MD.

Chandola, C., et al. (2006). Data Mining for Cyber Security, Book Chapter. In *Data Warehousing and Data Mining Techniques for Computer Security*, Editor Anoop Singhal, Springer.

Chen, W., Hsu, S., Shen, H. (2005). Application of SVM and ANN for Intrusion Detection. *Computers and Operations Research, 32(10),* 2617 – 2634.

Dao, V. & Vemuri, R. (2002). Computer Network Intrusion Detection: A Comparison of Neural Networks Methods, Differential Equations and Dynamical Systems. *Special Issue on Neural Networks*.

DeMeneses, Y. L., Roduit , P., Luisier, F., Jacot, J. (2005). Trajectory Analysis for Sport and Video Surveillance. *Electronic Letters on Computer Vision and Image Analysis*, 5(3), 148-156.

Dowell, C. & Ramstedt, P. (1990). The Computerwatch Data Reduction Tool, In Proceedings of the *13th National Computer Security Conference*, Washington, DC.

Eskin, E. (2000). Anomaly Detection over Noisy Data using Learned Probability Distributions, In Proceedings of the *International Conference on Machine Learning*, Stanford University, CA.

Eskin, E., et al. (2002). A Geometric Framework for Unsupervised Anomaly Detection: Detecting Intrusions in Unlabeled Data, in the book *Applications of Data Mining in Computer Security, Advances In Information Security*, Jajodia, S., Barbara, D. (Editors), Kluwer, Boston.

Fan, W., et al. (2001). Using Artificial Anomalies to Detect Unknown and Known Network Intrusions, In the Proceedings of the *First IEEE International conference on Data Mining*, San Jose, CA.

Ferrando, S., Gera, G., Regazzoni, C. (2006). Classification of Unattended and Stolen Objects in Video-Surveillance System. In Proceedings of the *IEEE International Conference on Video and Signal Based Surveillance*, Sydney, Australia.

Foresti, G. L., Piciarelli, C., Snidaro, L. (2005). Trajectory Clustering And Its Applications For Video Surveillance. In Proceedings of the *IEEE International Conference on Advanced Video and Signal based Surveillance*, 40-45, Como, Italy.

Guo, G.; Dyer, C.R.; Zhang, Z. (2005). Linear Combination Representation for Outlier Detection in Motion Tracking. In Proceedings of the *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Volume 2, 274 – 281.

Hawkins, S., He, H., Williams, G. & Baxter, R. (2002). Outlier Detection Using Replicator Neural Networks. In Proceedings of the *4th International Conference on Data Warehousing and Knowledge Discovery, Lecture Notes in Computer Science 2454*, Aix-en-Provence, France, 170-180.

Jones, D.A., Davis, C.E., Turnquist, M.A., Nozick, L.K. (2006). Physical Security and Vulnerability Modeling for Infrastructure Facilities. In Proceedings of the *39th Annual Hawaii International Conference on System Sciences*, 4 (04-07).

Joshi, M., Agarwal, R., Kumar, V. (2001). PNrule, Mining Needles in a Haystack: Classifying Rare Classes via Two-Phase Rule Induction, In Proceedings of the *ACM SIGMOD Conference on Management of Data*. Santa Barbara, CA.

Kim, D. S., Nguyen, H., Park, J. S. (2005). Genetic Algorithm to Improve SVM Based Network Intrusion Detection System, In Proceedings of the 19th International Conference on Advanced Information Networking and Applications, 155–158, Taipei, Taiwan.

Kumar, P., Devaraj, D. (2007). Network Intrusion Detection using Hybrid Neural Networks, In Proceedings of the *International Conference on Signal Processing, Communications and Networking*, Chennai, India, 563-569

Lazarevic, A., et al. (2003). A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection. In Proceedings of the *Third SIAM International Conference on Data Mining*, San Francisco, CA.

Lazarevic, A., Kumar, V. (2005a). Feature Bagging for Outlier Detection, In Proceedings of the *ACM-SIGKDD International Conference on Knowledge Discovery and Data Mining*, 157-166, Chicago, IL.

Lazarevic, A., Kumar, V. & Srivastava, J. (2005b). Intrusion Detection: A Survey. In the book *Managing Cyber Threats: Issues, Approaches and Challenges*. Kumar, V., Srivastava, J., Lazarevic, A. (Editors), Kluwer Academic Publishers.

Lee, W. & Stolfo, S.J. (2000). A Framework for Constructing Features and Models for Intrusion Detection Systems. *ACM Transactions on Information and System Security 3(4)*, 227-261.

Lee, W., Stolfo, S.J., Mok, K. (2001). Adaptive Intrusion Detection: A Data Mining Appproach. *Articial Intelligence Review*, 14, 533-567.

Leung, K., Leckie, C. (2005). Unsupervised Anomaly Detection in Network Intrusion Detection Using Clusters. In Proceedings of *the Twenty-eighth Australasian conference on Computer Science*, 333-342, Newcastle, Australia.

Li, W. (2004). Using Genetic Algorithm for Network Intrusion Detection, In Proceedings of *the US Department of Energy Cyber Security Group 2004 Training Conference*, Kansas City, KS.

Liepins, G. & Vaccaro, H. (1992). Intrusion Detection It's Role and Validation, *Computers and Security*, 347-355.

Lippmann, R.P., et al (1999). Results of the DARPA 1998 Offline Intrusion Detection Evaluation. In Proceedings of *Workshop on Recent Advances in Intrusion Detection*.

Lippmann, R., et al. (2000). The 1999 DARPA Off-Line Intrusion Detection Evaluation. *Computer Networks*.

Mahoney, M. & Chan, P. (2002). Learning Nonstationary Models of Normal Network Traffic for Detecting Novel Attacks. In Proceedings of the Eight *ACM International Conference on Knowledge Discovery and Data Mining*, Edmonton, Canada, 376-385.

Manganaris, S., et al. (2000). A Data Mining Analysis of RTID Alarms. *Computer Networks 34 (4)*, 571-577.

Mazu Networks (2007). Mazu Network Behavior Analysis (NBA) System. http://www.mazunetworks.com/products/mazu-nba.php.

Moore, D., et al. (2003). The Spread of the Sapphire/Slammer Worm. www.cs.berkeley.edu/~nweaver/sapphire.

Mukkamala, S., Sung, A. & Abraham, A. (2003). A Linear Genetic Programming Approach for Modeling Intrusion, In Proceedings of the *IEEE Congress on Evolutionary Computation*, Perth, Australia.

Naftel, A., Khalid, S. (2006). Classifying Spatiotemporal Object Trajectories Using Unsupervised Learning in the Coefficient Feature Space. *Multimedia Systems*, Vol. 12, 227–238.

Pokrajac, D. Lazarevic, A., Latecki, L. (2007). Incremental Local Outlier Detection for Data Streams. In Proceedings of the *IEEE Symposium on Computational Intelligence and Data Mining*, Honolulu, HI.

Schultz, M., Eskin, E., Zadok, E. & Stolfo, S. (2001). Data Mining Methods for Detection of New Malicious Executables. In Proceedings of the IEEE Symposium on Security and Privacy, Oakland, CA, 38-49.

Seifert, J. (2007). Data Mining and Homeland Security, An Overview, CRS Report for Congress, http://www.fas.org/sgp/crs/intel/RL31798.pdf

Vlachos, M., Gunopoulos, D., Kollios, G. (2002). Discovering Similar Multidimensional Trajectories. In Proceedings of the *18th International Conference on Data Engineering*, 673-683, San Jose, CA.

Wang, K., Stolfo, S. (2004). Anomalous Payload-based Network Intrusion Detection. In Proceedings of *the*

Recent Advances in Intrusion Detection Symposium, 203-222, Sophia Antipolis, France.

Yamanishi, K., Takeuchi, J., Williams G. & Milne, P. (2000). On-line Unsupervised Oultlier Detection Using Finite Mixtures with Discounting Learning Algorithms, In Proceedings of the *Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Boston, MA, 320-324.

Yanagisawa, Y., Satoh, T. (2006). Clustering Multidimensional Trajectories based on Shape and Velocity. In Proceedings of the *22nd International Conference on Data Engineering*, Atlanta, GA.

Ye, N. & Chen, Q. (2001a). An Anomaly Detection Technique Based on a Chi-Square Statistic for Detecting Intrusions Into Information Systems. *Quality and Reliability Engineering International Journal*, *17(2)*, 105-112.

Ye, N. & Li, X. (June 2001b). A Scalable Clustering Technique for Intrusion Signature Recognition. In Proceedings of the *IEEE Workshop on Information Assurance and Security*, US Military Academy, West Point, NY.

Zanero, S. (2006). Unsupervised Learning Algorithms for Intrusion Detection, Ph.D. Thesis, DEI Politecnico di Milano, Italy.

Zhang, C., Jiang, J. & Kamel, M. (2005). Intrusion detection using hierarchical neural networks, *Pattern Recognition Letters,26*(*6*), 779-791.

KEY TERMS

Intrusion: Malicious, externally induced, operational fault in the computer system.

Intrusion Detection: Identifying a set of malicious actions that compromise the integrity, confidentiality, and availability of information resources.

Misuse Detection: Analysis strategy that looks for events or sets of events that match a predefined pattern of a known attack.

Anomaly Detection: Analysis strategy that identifies intrusions as unusual behavior that differs from the normal behavior of the monitored system.

Signature-based Intrusion Detection: Analysis strategy where monitored events are matched against a database of attack signatures to detect intrusions.

Worms: Self-replicating programs that aggressively spread through a network, by taking advantage of automatic packet sending and receiving features found on many computers.

Physical / Electronic Security System: System consisting of tens or hundred of sensors (e,g., passive infrared, magnetic contacts, acoustic glass break, video cameras) that prevent, detect or deter attackers from accessing a facility, resource, or information stored on physical media.

Video Surveillance: Use of video cameras to transmit a signal to a specific, limited set of monitors where it is used for inspection of safety, possible crimes, traffic, etc.

Data Mining in the Telecommunications Industry

Gary Weiss
Fordham University, USA

INTRODUCTION

The telecommunications industry was one of the first to adopt data mining technology. This is most likely because telecommunication companies routinely generate and store enormous amounts of high-quality data, have a very large customer base, and operate in a rapidly changing and highly competitive environment. Telecommunication companies utilize data mining to improve their marketing efforts, identify fraud, and better manage their telecommunication networks. However, these companies also face a number of data mining challenges due to the enormous size of their data sets, the sequential and temporal aspects of their data, and the need to predict very rare events—such as customer fraud and network failures—in real-time.

The popularity of data mining in the telecommunications industry can be viewed as an extension of the use of expert systems in the telecommunications industry (Liebowitz, 1988). These systems were developed to address the complexity associated with maintaining a huge network infrastructure and the need to maximize network reliability while minimizing labor costs. The problem with these expert systems is that they are expensive to develop because it is both difficult and time-consuming to elicit the requisite domain knowledge from experts. Data mining can be viewed as a means of automatically generating some of this knowledge directly from the data.

BACKGROUND

The data mining applications for any industry depend on two factors: the data that are available and the business problems facing the industry. This section provides background information about the data maintained by telecommunications companies. The challenges associated with mining telecommunication data are also described in this section.

Telecommunication companies maintain data about the phone calls that traverse their networks in the form of *call detail* records, which contain descriptive information for each phone call. In 2001, AT&T long distance customers generated over 300 million call detail records per day (Cortes & Pregibon, 2001) and, because call detail records are kept online for several months, this meant that billions of call detail records were readily available for data mining. Call detail data is useful for marketing and fraud detection applications.

Telecommunication companies also maintain extensive customer information, such as billing information, as well as information obtained from outside parties, such as credit score information. This information can be quite useful and often is combined with telecommunication-specific data to improve the results of data mining. For example, while call detail data can be used to identify suspicious calling patterns, a customer's credit score is often incorporated into the analysis before determining the likelihood that fraud is actually taking place.

Telecommunications companies also generate and store an extensive amount of data related to the operation of their networks. This is because the network elements in these large telecommunication networks have some self-diagnostic capabilities that permit them to generate both status and alarm messages. These streams of messages can be mined in order to support network management functions, namely fault isolation and prediction.

The telecommunication industry faces a number of data mining challenges. According to a Winter Corporation survey (2003), the three largest databases all belong to telecommunication companies, with France Telecom, AT&T, and SBC having databases with 29, 26, and 25 Terabytes, respectively. Thus, the scalability of data mining methods is a key concern. A second issue is that telecommunication data is often in the form of transactions/events and is not at the proper semantic level for data mining. For example, one typically wants to mine call detail data at the customer (i.e., phone-

line) level but the raw data represents individual phone calls. Thus it is often necessary to *aggregate* data to the appropriate semantic level (Sasisekharan, Seshadri & Weiss, 1996) before mining the data. An alternative is to utilize a data mining method that can operate on the transactional data directly and extract sequential or temporal patterns (Klemettinen, Mannila & Toivonen, 1999; Weiss & Hirsh, 1998).

Another issue arises because much of the telecommunications data is generated in real-time and many telecommunication applications, such as fraud identification and network fault detection, need to *operate* in real-time. Because of its efforts to address this issue, the telecommunications industry has been a leader in the research area of mining data streams (Aggarwal, 2007). One way to handle data streams is to maintain a *signature* of the data, which is a summary description of the data that can be updated quickly and incrementally. Cortes and Pregibon (2001) developed signature-based methods and applied them to data streams of call detail records. A final issue with telecommunication data and the associated applications involves rarity. For example, both telecommunication fraud and network equipment failures are relatively rare. Predicting and identifying rare events has been shown to be quite difficult for many data mining algorithms (Weiss, 2004) and therefore this issue must be handled carefully in order to ensure reasonably good results.

MAIN FOCUS

Numerous data mining applications have been deployed in the telecommunications industry. However, most applications fall into one of the following three categories: marketing, fraud detection, and network fault isolation and prediction.

Telecommunications Marketing

Telecommunication companies maintain an enormous amount of information about their customers and, due to an extremely competitive environment, have great motivation for exploiting this information. For these reasons the telecommunications industry has been a leader in the use of data mining to identify customers, retain customers, and maximize the profit obtained from each customer. Perhaps the most famous use of data mining to acquire new telecommunications customers was MCI's Friends and Family program. This program, long since retired, began after marketing researchers identified many small but well connected subgraphs in the graphs of calling activity (Han, Altman, Kumar, Mannila & Pregibon, 2002). By offering reduced rates to customers in one's calling circle, this marketing strategy enabled the company to use their own customers as salesmen. This work can be considered an early use of social-network analysis and link mining (Getoor & Diehl, 2005). A more recent example uses the interactions between consumers to identify those customers likely to adopt new telecommunication services (Hill, Provost & Volinsky, 2006). A more traditional approach involves generating customer profiles (i.e., signatures) from call detail records and then mining these profiles for marketing purposes. This approach has been used to identify whether a phone line is being used for voice or fax (Kaplan, Strauss & Szegedy, 1999) and to classify a phone line as belonging to a either business or residential customer (Cortes & Pregibon, 1998).

Over the past few years, the emphasis of marketing applications in the telecommunications industry has shifted from identifying new customers to measuring customer value and then taking steps to retain the most profitable customers. This shift has occurred because it is much more expensive to acquire new telecommunication customers than retain existing ones. Thus it is useful to know the *total lifetime value* of a customer, which is the total net income a company can expect from that customer over time. A variety of data mining methods are being used to model customer lifetime value for telecommunication customers (Rosset, Neumann, Eick & Vatnik, 2003; Freeman & Melli, 2006).

A key component of modeling a telecommunication customer's value is estimating how long they will remain with their current carrier. This problem is of interest in its own right since if a company can predict when a customer is likely to leave, it can take proactive steps to retain the customer. The process of a customer leaving a company is referred to as *churn*, and *churn analysis* involves building a model of customer attrition. Customer churn is a huge issue in the telecommunication industry where, until recently, telecommunication companies routinely offered large cash incentives for customers to switch carriers. Numerous systems and methods have been developed to predict customer churn (Wei & Chin, 2002; Mozer, Wolniewicz, Grimes, Johnson, & Kaushansky, 2000; Mani, Drew, Betz & Datta, 1999; Masand, Datta, Mani,

& Li, 1999). These systems almost always utilize call detail and contract data, but also often use other data about the customer (credit score, complaint history, etc.) in order to improve performance. Churn prediction is fundamentally a very difficult problem and, consequently, systems for predicting churn have been only moderately effective—only demonstrating the ability to identify some of the customers most likely to churn (Masand et al., 1999).

Telecommunications Fraud Detection

Fraud is very serious problem for telecommunication companies, resulting in billions of dollars of lost revenue each year. Fraud can be divided into two categories: subscription fraud and superimposition fraud (Fawcett and Provost, 2002). *Subscription fraud* occurs when a customer opens an account with the intention of never paying the account and *superimposition fraud* occurs when a perpetrator gains illicit access to the account of a legitimate customer. In this latter case, the fraudulent behavior will often occur in parallel with legitimate customer behavior (i.e., is superimposed on it). Superimposition fraud has been a much more significant problem for telecommunication companies than subscription fraud. Ideally, both subscription fraud and superimposition fraud should be detected immediately and the associated customer account deactivated or suspended. However, because it is often difficult to distinguish between legitimate and illicit use with limited data, it is not always feasible to detect fraud as soon as it begins. This problem is compounded by the fact that there are substantial costs associated with investigating fraud, as well as costs if usage is mistakenly classified as fraudulent (e.g., an annoyed customer).

The most common technique for identifying superimposition fraud is to compare the customer's current calling behavior with a profile of his past usage, using deviation detection and anomaly detection techniques. The profile must be able to be quickly updated because of the volume of call detail records and the need to identify fraud in a timely manner. Cortes and Pregibon (2001) generated a signature from a data stream of call-detail records to concisely describe the calling behavior of customers and then they used anomaly detection to "measure the unusualness of a new call relative to a particular account." Because new behavior does not necessarily imply fraud, this basic approach was augmented by comparing the new calling behavior to profiles of generic fraud—and fraud is only signaled if the behavior matches one of these profiles. Customer level data can also aid in identifying fraud. For example, price plan and credit rating information can be incorporated into the fraud analysis (Rosset, Murad, Neumann, Idan, & Pinkas, 1999). More recent work using signatures has employed dynamic clustering as well as deviation detection to detect fraud (Alves et al., 2006). In this work, each signature was placed within a cluster and a change in cluster membership was viewed as a potential indicator of fraud.

There are some methods for identifying fraud that do not involve comparing new behavior against a profile of old behavior. Perpetrators of fraud rarely work alone. For example, perpetrators of fraud often act as brokers and sell illicit service to others—and the illegal buyers will often use different accounts to call the same phone number again and again. Cortes and Pregibon (2001) exploited this behavior by recognizing that certain phone numbers are repeatedly called from compromised accounts and that calls to these numbers are a strong indicator that the current account may be compromised. A final method for detecting fraud exploits human pattern recognition skills. Cox, Eick & Wills (1997) built a suite of tools for visualizing data that was tailored to show calling activity in such a way that unusual patterns are easily detected by users. These tools were then used to identify international calling fraud.

Telecommunication Network Fault Isolation and Prediction

Monitoring and maintaining telecommunication networks is an important task. As these networks became increasingly complex, expert systems were developed to handle the alarms generated by the network elements (Weiss, Ros & Singhal, 1998). However, because these systems are expensive to develop and keep current, data mining applications have been developed to identify and predict network faults. Fault identification can be quite difficult because a single fault may result in a cascade of alarms—many of which are not associated with the root cause of the problem. Thus an important part of fault identification is alarm correlation, which enables multiple alarms to be recognized as being related to a single fault.

The Telecommunication Alarm Sequence Analyzer (TASA) is a data mining tool that aids with fault identification by looking for frequently occurring temporal patterns of alarms (Klemettinen, Mannila & Toivonen, 1999). Patterns detected by this tool were then used to help construct a rule-based alarm correlation system. Another effort, used to predict telecommunication switch failures, employed a genetic algorithm to mine historical alarm logs looking for predictive sequential and temporal patterns (Weiss & Hirsh, 1998). One limitation with the approaches just described is that they ignore the structural information about the underlying network. The quality of the mined sequences can be improved if topological proximity constraints are considered in the data mining process (Devitt, Duffin and Moloney, 2005) or if substructures in the telecommunication data can be identified and exploited to allow simpler, more useful, patterns to be learned (Baritchi, Cook, & Lawrence, 2000). Another approach is to use Bayesian Belief Networks to identify faults, since they can reason about causes and effects (Sterritt, Adamson, Shapcott & Curran, 2000).

FUTURE TRENDS

Data mining should play an important and increasing role in the telecommunications industry due to the large amounts of high quality data available, the competitive nature of the industry and the advances being made in data mining. In particular, advances in mining data streams, mining sequential and temporal data, and predicting/classifying rare events should benefit the telecommunications industry. As these and other advances are made, more reliance will be placed on the knowledge acquired through data mining and less on the knowledge acquired through the time-intensive process of eliciting domain knowledge from experts—although we expect human experts will continue to play an important role for some time to come.

Changes in the nature of the telecommunications industry will also lead to the development of new applications and the demise of some current applications. As an example, the main application of fraud detection in the telecommunications industry used to be in cellular cloning fraud, but this is no longer the case because the problem has been largely eliminated due to technological advances in the cell phone authentication process. It is difficult to predict what future changes will face the telecommunications industry, but as telecommunication companies start providing television service to the home and more sophisticated cell phone services become available (e.g., music, video, etc.), it is clear that new data mining applications, such as recommender systems, will be developed and deployed.

Unfortunately, there is also one troubling trend that has developed in recent years. This concerns the increasing belief that U.S. telecommunication companies are too readily sharing customer records with governmental agencies. This concern arose in 2006 due to revelations—made public in numerous newspaper and magazine articles—that telecommunications companies were turning over information on calling patterns to the National Security Agency (NSA) for purposes of data mining (Krikke, 2006). If this concern continues to grow unchecked, it could lead to restrictions that limit the use of data mining for legitimate purposes.

CONCLUSION

The telecommunications industry has been one of the early adopters of data mining and has deployed numerous data mining applications. The primary applications relate to marketing, fraud detection, and network monitoring. Data mining in the telecommunications industry faces several challenges, due to the size of the data sets, the sequential and temporal nature of the data, and the real-time requirements of many of the applications. New methods have been developed and existing methods have been enhanced to respond to these challenges. The competitive and changing nature of the industry, combined with the fact that the industry generates enormous amounts of data, ensures that data mining will play an important role in the future of the telecommunications industry.

REFERENCES

Aggarwal, C. (Ed.). (2007). *Data Streams: Models and Algorithms*. New York: Springer.

Alves, R., Ferreira, P., Belo, O., Lopes, J., Ribeiro, J., Cortesao, L., & Martins, F. (2006). Discovering telecom fraud situations through mining anomalous behavior patterns. *Proceedings of the ACM SIGKDD Workshop on Data Mining for Business Applications* (pp. 1-7). New York: ACM Press.

Baritchi, A., Cook, D., & Holder, L. (2000). Discovering structural patterns in telecommunications data. *Proceedings of the Thirteenth Annual Florida AI Research Symposium* (pp. 82-85).

Cortes, C., & Pregibon, D (1998). Giga-mining. *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining* (pp. 174-178). New York, NY: AAAI Press.

Cortes, C., & Pregibon, D. (2001). Signature-based methods for data streams. *Data Mining and Knowledge Discovery,* 5(3), 167-182.

Cox, K., Eick, S., & Wills, G. (1997). Visual data mining: Recognizing telephone calling fraud. *Data Mining and Knowledge Discovery*, 1(2), 225-231.

Devitt, A., Duffin, J., & Moloney, R. (2005). Topographical proximity for mining network alarm data. *Proceedings of the 2005 ACM SIGCOMM Workshop on Mining Network Data* (pp. 179-184). New York: ACM Press.

Fawcett, T., & Provost, F. (2002). Fraud Detection. In W. Klosgen & J. Zytkow (Eds.), *Handbook of Data Mining and Knowledge Discovery* (pp. 726-731). New York: Oxford University Press.

Freeman, E., & Melli, G. (2006). Championing of an LTV model at LTC. *SIGKDD Explorations*, 8(1), 27-32.

Getoor, L., & Diehl, C.P. (2005). Link mining: A survey. *SIGKDD Explorations*, 7(2), 3-12.

Hill, S., Provost, F., & Volinsky, C. (2006). Network-based marketing: Identifying likely adopters via consumer networks. *Statistical Science*, 21(2), 256-276.

Kaplan, H., Strauss, M., & Szegedy, M. (1999). Just the fax—differentiating voice and fax phone lines using call billing data. *Proceedings of the Tenth Annual ACM-SIAM Symposium on Discrete Algorithms* (pp. 935-936). Philadelphia, PA: Society for Industrial and Applied Mathematics.

Klemettinen, M., Mannila, H., & Toivonen, H. (1999). Rule discovery in telecommunication alarm data. *Journal of Network and Systems Management,* 7(4), 395-423.

Krikke, J. (2006). Intelligent surveillance empowers security analysts. *IEEE Intelligent Systems*, 21(3), 102-104.

Liebowitz, J. (1988). *Expert System Applications to Telecommunications*. New York, NY: John Wiley & Sons.

Mani, D., Drew, J., Betz, A., & Datta, P (1999). Statistics and data mining techniques for lifetime value modeling. *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 94-103). New York, NY: ACM Press.

Masand, B., Datta, P., Mani, D., & Li, B. (1999). CHAMP: A prototype for automated cellular churn prediction. *Data Mining and Knowledge Discovery*, 3(2), 219-225.

Mozer, M., Wolniewicz, R., Grimes, D., Johnson, E., & Kaushansky, H. (2000). Predicting subscriber dissatisfaction and improving retention in the wireless telecommunication industry. *IEEE Transactions on Neural Networks*, 11, 690-696.

Rosset, S., Murad, U., Neumann, E., Idan, Y., & Gadi, P. (1999). Discovery of fraud rules for telecommunications—challenges and solutions. *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 409-413). New York: ACM Press.

Rosset, S., Neumann, E., Eick, U., & Vatnik (2003). Customer lifetime value models for decision support. *Data Mining and Knowledge Discovery*, 7(3), 321-339.

Sasisekharan, R., Seshadri, V., & Weiss, S (1996). Data mining and forecasting in large-scale telecommunication networks. *IEEE Expert*, 11(1), 37-43.

Sterritt, R., Adamson, K., Shapcott, C., & Curran, E. (2000). Parallel data mining of Bayesian networks from telecommunication network data. *Proceedings of the 14th International Parallel and Distributed Processing Symposium*, IEEE Computer Society.

Wei, C., & Chiu, I (2002). Turning telecommunications call details to churn prediction: A data mining approach. *Expert Systems with Applications*, 23(2), 103-112.

Weiss, G., & Hirsh, H (1998). Learning to predict rare events in event sequences. In R. Agrawal & P. Stolorz (Eds.), *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining* (pp. 359-363). Menlo Park, CA: AAAI Press.

Weiss, G., Ros, J., & Singhal, A. (1998). ANSWER: Network monitoring using object-oriented rule. *Proceedings of the Tenth Conference on Innovative Applications of Artificial Intelligence* (pp. 1087-1093). Menlo Park: AAAI Press.

Weiss, G. (2004). Mining with rarity: A unifying framework. *SIGKDD Explorations*, 6(1), 7-19.

Winter Corporation (2003). *2003 Top 10 Award Winners*. Retrieved October 8, 2005, from http://www.wintercorp.com/VLDB/2003_TopTen_Survey/TopTenwinners.asp

KEY TERMS

Bayesian Belief Network: A model that predicts the probability of events or conditions based on causal relations.

Call Detail Record: Contains the descriptive information associated with a single phone call.

Churn: Customer attrition. Churn prediction refers to the ability to predict that a customer will leave a company before the change actually occurs.

Signature: A summary description of a subset of data from a data stream that can be updated incrementally and quickly.

Subscription Fraud: Occurs when a perpetrator opens an account with no intention of ever paying for the services incurred.

Superimposition Fraud: Occurs when a perpetrator gains illicit access to an account being used by a legitimate customer and where fraudulent use is "superimposed" on top of legitimate use.

Total Lifetime Value: The total net income a company can expect from a customer over time.

Data Mining Lessons Learned in the Federal Government

Les Pang
National Defense University, USA

INTRODUCTION

Data mining has been a successful approach for improving the level of business intelligence and knowledge management throughout an organization. This article identifies lessons learned from data mining projects within the federal government including military services. These lessons learned were derived from the following project experiences:

- Defense Medical Logistics Support System Data Warehouse Program
- Department of Defense (DoD) Defense Financial and Accounting Service (DFAS) "Operation Mongoose"
- DoD Computerized Executive Information System (CEIS)
- Department of Homeland Security's Analysis, Dissemination, Visualization, Insight, and Semantic Enhancement (ADVISE) Program
- Department of Transportation (DOT) Executive Reporting Framework System
- Federal Aviation Administration (FAA) Aircraft Accident Data Mining Project
- General Accountability Office (GAO) Data Mining of DoD Purchase and Travel Card Programs
- U.S. Coast Guard Executive Information System
- Veteran Administrations (VA) Demographics System

BACKGROUND

Data mining involves analyzing diverse data sources in order to identify relationships, trends, deviations and other relevant information that would be valuable to an organization. This approach typically examines large single databases or linked databases that are dispersed throughout an organization. Pattern recognition technologies and statistical and mathematical techniques are often used to perform data mining. By utilizing this approach, an organization can gain a new level of corporate knowledge that can be used to address its business requirements.

Many agencies in the federal government have applied a data mining strategy with significant success. This chapter aims to identify the lessons gained as a result of these many data mining implementations within the federal sector. Based on a thorough literature review, these lessons were uncovered and selected by the author as being critical factors which led toward the success of the real-world data mining projects. Also, some of these lessons reflect novel and imaginative practices.

MAIN THRUST

Each lesson learned (indicated in **boldface**) is listed below. Following each practice is a description of illustrative project or projects (indicated in *italics*), which support the lesson learned.

Avoid the Privacy Trap

DoD Computerized Executive Information System: Patients as well as the system developers indicate their concern for protecting the privacy of individuals -- their medical records need safeguards. "Any kind of large database like that where you talk about personal info raises red flags," said Alex Fowler, a spokesman for the Electronic Frontier Foundation. "There are all kinds of questions raised about who accesses that info or protects it and how somebody fixes mistakes" (Hamblen, 1998).

Proper security safeguards need to be implemented to protect the privacy of those in the mined databases. Vigilant measures are needed to ensure that only authorized individuals have the capability of accessing, viewing and analyzing the data. Efforts should also

be made to protect the data through encryption and identity management controls.

Evidence of the public's high concern for privacy was the demise of the Pentagon's $54 million Terrorist Information Awareness (originally, Total Information Awareness) effort -- the program in which government computers were to be used to scan an enormous array of databases for clues and patterns related to criminal or terrorist activity. To the dismay of privacy advocates, many government agencies are still mining numerous databases (General Accounting Office, 2004; Gillmor, 2004). "Data mining can be a useful tool for the government, but safeguards should be put in place to ensure that information is not abused," stated the chief privacy officer for the Department of Homeland Security (Sullivan, 2004). Congressional concerns on privacy are so high that the body is looking at introducing legislation that would require agencies to report to Congress on data mining activities to support homeland security purposes (Miller, 2004). Privacy advocates have also expressed concern over Analysis, Dissemination, Visualization, Insight, and Semantic Enhancement (ADVISE), a data mining research and development program within the Department of Homeland Security (DHS). (Clayton, 2006).

The Center for Democracy and Technology calls for three technical solutions to address privacy: apply anonymization techniques so that data mining analysts can share information with authorities without disclosing the identity of individuals; include authorization requirements into government systems for reviewing data to ensure that only those who need to see the data do, and utilize audit logs to identify and track inappropriate access to data sources.

Steer Clear of the "Guns Drawn" Mentality if Data Mining Unearths a Discovery

DoD Defense Finance & Accounting Service's Operation Mongoose was a program aimed to discover billing errors and fraud through data mining. About 2.5 million financial transactions were searched to locate inaccurate charges. This approach detected data patterns that might indicate improper use. Examples include purchases made on weekends and holidays, entertainment expenses, highly frequent purchases, multiple purchases from a single vendor and other transactions that do not match with the agency's past purchasing patterns. It turned up a cluster of 345 cardholders (out of 400,000) who had made suspicious purchases.

However, the process needs some fine-tuning. As an example, buying golf equipment appeared suspicious until it was learned that a manager of a military recreation center had the authority to buy the equipment. Also, casino-related expense revealed to be a commonplace hotel bill. Nevertheless, the data mining results have shown sufficient potential that data mining will become a standard part of the Department's efforts to curb fraud.

Create a Business Case Based on Case Histories to Justify Costs

FAA Aircraft Accident Data Mining Project involved the Federal Aviation Administration hiring MITRE Corporation to identify approaches it can use to mine volumes of aircraft accident data to detect clues about their causes and how those clues could help avert future crashes (Bloedorn, 2000). One significant data mining finding was that planes with instrument displays that can be viewed without requiring a pilot to look away from the windshield were damaged a smaller amount in runway accidents than planes without this feature.

On the other hand, the government is careful about committing significant funds to data mining projects. "One of the problems is how do you prove that you kept the plane from falling out of the sky," said Trish Carbone, a technology manager at MITRE. It is difficult to justify data mining costs and relate it to benefits (Matthews, 2000).

One way to justify data mining program is to look at past successes in data mining. Historically, fraud detection has been the highest payoff in data mining, but other areas have also benefited from the approach such as in sales and marketing in the private sector. Statistics (dollars recovered) from efforts such as this can be used to support future data mining projects.

Use Data Mining for Supporting Budgetary Requests

Veteran's Administration Demographics System predicts demographic changes based on patterns among its 3.6 million patients as well as data gathered from insurance companies. Data mining enables the VA to provide Congress with much more accurate budget requests. The VA spends approximately $19 billion a

year to provide medical care to veterans. All government agencies such as the VA are under increasing scrutiny to prove that they are operating effectively and efficiently. This is particularly true as a driving force behind the President's Management Agenda (Executive Office of the President, 2002). For many, data mining is becoming the tool of choice to highlight good performance or dig out waste.

United States Coast Guard developed an executive information system designed for managers to see what resources are available to them and better understand the organization's needs. Also, it is also used to identify relationships between Coast Guard initiatives and seizures of contraband cocaine and establish tradeoffs between costs of alternative strategies. The Coast Guard has numerous databases; content overlap each other; and only one employee understands each database well enough to extract information. In addition, field offices are organized geographically but budgets are drawn up by programs that operate nationwide. Therefore, there is a disconnect between organizational structure and appropriations structure. The Coast Guard successfully used their data mining program to overcome these issues (Ferris, 2000).

DOT Executive Reporting Framework (ERF) aims to provide complete, reliable and timely information in an environment that allows for the cross-cutting identification, analysis, discussion and resolution of issues. The ERF also manages grants and operations data. Grants data shows the taxes and fees that DOT distributes to the states' highway and bridge construction, airport development and transit systems. Operations data covers payroll, administrative expenses, travel, training and other operations cost. The ERF system accesses data from various financial and programmatic systems in use by operating administrators.

Before 1993 there was no financial analysis system to compare the department's budget with congressional appropriations. There was also no system to track performance against the budget or how they were doing with the budget. The ERF system changed this by tracking of the budget and providing the ability to correct any areas that have been over planned. Using ERF, adjustments were made within the quarter so the agency did not go over budget. ERF is being extended to manage budget projections, development and formulation. It can be used as a proactive tool, which allows an agency to be more dynamically to project ahead. The system has improved financial accountability in the agency (Ferris, 1999).

Give Users Continual Computer-Based Training

DoD Medical Logistics Support System (DMLSS) built a data warehouse with a front-end decision support/data mining tools to help manage the growing costs of health care, enhance health care delivery, enhance health delivery in peacetime and promote wartime readiness and sustainability. DMLSS is responsible for the supply of medical equipment and medicine worldwide for DoD medical care facilities. The system received recognition for reducing the inventory in its medical depot system by 80 percent and reducing supply request response time from 71 to 15 days (Government Computer News, 2001). One major challenge faced by the agency was the difficulty in keeping up on the training of users because of the constant turnover of military personnel. It was determined that there is a need to provide quality computer-based training on a continuous basis (Olsen, 1997).

Provide the Right Blend of Technology, Human Capital Expertise and Data Security Measures

General Accountability Office used data mining to identify numerous instances of illegal purchases of goods and services from restaurants, grocery stores, casinos, toy stores, clothing retailers, electronics stores, gentlemen's clubs, brothels, auto dealers and gasoline service stations. This was all part of their effort to audit and investigate federal government purchase and travel card and related programs (General Accounting Office, 2003).

Data mining goes beyond using the most effective technology and tools. There must be well-trained individuals involved who know about the process, procedures and culture of the system being investigated. They need to understand the capabilities and limitation of data mining concepts and tools. In addition, these individual must recognize the data security issues associated with the use of large, complex and detailed databases.

Leverage Data Mining to Detect Identity Theft

Identity theft, the illegal use of another individual's identity, has resulted in significant financial losses for

innocent individuals. *Department of Education's Title IV Identity Theft Initiative* deals with identity theft focusing on education loans. (General Accounting Office, 2004)

FUTURE TRENDS

Despite the privacy concerns, data mining continues to offer much potential in identifying waste and abuse, potential terrorist and criminal activity, and identify clues to improve efficiency and effectiveness within organizations. This approach will become more pervasive because of its integration with online analytical tools, the improved ease of use in utilizing data mining tools and the appearance of novel visualization techniques for reporting results. Also, the emergence of a new branch of data mining called text mining to help improve the efficiency of searching on the Web. This approach transforms textual data into a useable format that facilitates classifying documents, finds explicit relationships or associations among documents, and clusters documents into categories (SAS, 2004).

CONCLUSION

These lessons learned may or may not fit in all environments due to cultural, social and financial considerations. However, the careful review and selection of relevant lessons learned could result in addressing the required goals of the organization by improving the level of corporate knowledge.

A decision maker needs to think "outside the box" and move away from the traditional approaches to successfully implement and manage their programs. Data mining poses a challenging but highly effective approach to improve business intelligence within one's domain.

ACKNOWLEDGMENT

The views expressed in this article are those of the author and do not reflect the official policy or position of the National Defense University, the Department of Defense or the U.S. Government.

REFERENCES

Bloedorn, E. (2000). *Data mining for aviation safety*. MITRE Publications.

Clayton, M., & P. Rosenzweig (2006). *US Plans Massive Data Sweep.*

Dempsey, J. (2004). *Technologies that can protect privacy that can protect privacy as information is shared to combat terrorism.* Center for Democracy and Technology.

Executive Office of the President, Office of Management and Budget. (2002). *President's management agenda*. Fiscal Year 2002.

Ferris, N. (1999). 9 Hot Trends for '99. *Government Executive.*

Ferris, N. (2000). Information is power. *Government Executive.*

General Accounting Office. (2003). *Data mining: Results and challenges for government program audits and investigations*. GAO-03-591T.

General Accounting Office. (2004). *Data mining: Federal efforts cover a wide range of uses*. GAO-04-584.

Gillmor, D. (2004). Data mining by government rampant. *eJournal*.

Government Computer News. (2001). Ten agencies honored for innovative projects. *Government Computer News.*

Hamblen, M. (1998). Pentagon to deploy huge medical data warehouse. *Computer World.*

Matthews, W. (2000). Digging Digital Gold. *Federal Computer Week.*

Miller, J. (2004). Lawmakers renew push for data-mining law. *Government Computer News.*

Olsen, F. (1997). Health record project hits pay dirt. *Government Computer News.*

SAS. (2004). *SAS Text Miner*.

Schwartz, A. (2000). Making the Web safe. *Federal Computer Week.*

Sullivan, A. (2004). *U.S. Government still data mining*. Reuters.

KEY TERMS

Computer-Based Training: A recent approach involving the use of microcomputer, optical disks such as compact disks, and/or the Internet to address an organization's training needs.

Executive Information sSystem: An application designed for the top executives that often features a dashboard interface, drill-down capabilities and trend analysis.

Federal Government: The national government of the United States, established by the Constitution, which consists of the executive, legislative, and judicial branches. The head of the executive branch is the President of the United States. The legislative branch consists of the United States Congress, and the Supreme Court of the United States is the head of the judicial branch.

Legacy System: Typically, a database management system in which an organization has invested considerable time and money and resides on a mainframe or minicomputer.

Logistics Support System: A computer package that assists in the planning and deploying the movement and maintenance of forces in the military. The package could deals with the design and development, acquisition, storage, movement, distribution, maintenance, evacuation and disposition of material; movement, evacuation, and hospitalization of personnel; acquisition of construction, maintenance, operation and disposition of facilities; and acquisition of furnishing of services.

Purchase Cards: Credit cards used in the federal government used by authorized government official for small purchases, usually under $2,500.

Travel Cards: Credit cards issued to federal employees to pay for costs incurred on official business travel.

A Data Mining Methodology for Product Family Design

Seung Ki Moon
The Pennsylvania State University, USA

Timothy W. Simpson
The Pennsylvania State University, USA

Soundar R. T. Kumara
The Pennsylvania State University, USA

INTRODUCTION

Many companies strive to maximize resource utilization by sharing and reusing distributed design knowledge and information when developing new products. By sharing and reusing assets such as components, modules, processes, information, and knowledge across a family of products and services, companies can efficiently develop a set of differentiated products by improving the flexibility and responsiveness of product development (Simpson, 2004). Product family planning is a way to achieve cost-effective mass customization by allowing highly differentiated products to be developed from a shared platform while targeting products to distinct market segments (Shooter et al., 2005).

In product design, data mining can be used to help identify customer needs, to find relationships between customer needs and functional requirements, and to cluster products based on functional similarity to facilitate modular design (Braha, 2001). The objective in this chapter is to introduce a methodology for identifying a platform along with variant and unique modules in a product family using design knowledge extracted with data mining techniques. During conceptual design, data mining can facilitate decision-making when selecting design concepts by extracting design knowledge and rules, clustering design cases, and exploring conceptual designs in large product design databases interactively (Braha, 2001). Moreover, since design knowledge for a product depends on the experience and knowledge of designers, representation of design knowledge, such as linguistic representation, may fail to describe a crisp representation completely. When clustering design knowledge, the knowledge is needed to assign to clusters with varying degrees of membership. Fuzzy membership can be used to represent and model the fuzziness of design knowledge (Braha, 2001). Design knowledge can be defined as linguistic variables based on the fuzzy set theory to support decision-making in product development (Ma et al., 2007).

BACKGROUND

A product family is a group of related products based on a product platform, facilitating mass customization by providing a variety of products for different market segments cost-effectively (Simpson et al., 2005). A successful product family depends on how well the trade-offs between the economic benefits and performance losses incurred from having a platform are managed. Various data mining approaches have been applied to product family design and product development. Clustering can be used to group customers or functions of similar behavior (Agard & Kusiak, 2004; Jiao & Zhang, 2005). Also, functional requirements in existing products can be clustered based on the similarity between them. This process can be achieved by using clustering methods such as the k-means algorithm, hierarchical algorithms, pattern recognition, Bayesian statistics, neural networks, and support vector machines. Agard and Kusiak (2004) proposed a three-step method for the design of product families based on the analysis of customers' requirements using a data mining approach. In the first step, data mining algorithms are used for customer segmentation. The second step provides a function structure to satisfy the diversified requirements. A product structure and distinguished modules for the product variability are designed in the final step. Moon et al. (2006) introduced a methodology

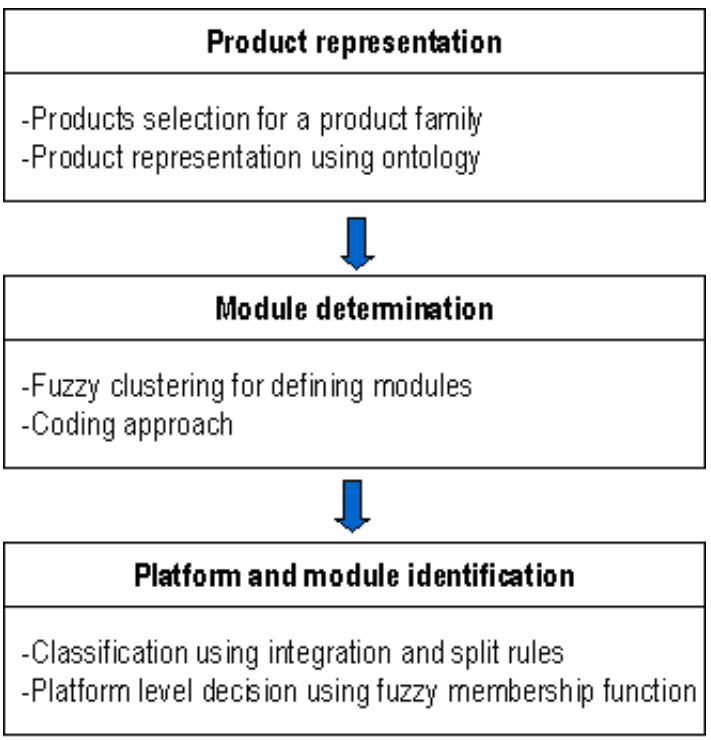
for identifying a platform and modules for product family design using fuzzy clustering, association rule mining, and classification. Ma et al. (2007) presented a decision-making support model for customized product color combination using the fuzzy analytic hierarchy process (FAHP) that utilizes the fuzzy set theory to integrate with AHP.

Sharing and reusing product design information can help eliminate such wastes and facilitate good product family design. To share and reuse the information, it is important to adopt an appropriate representation scheme for components and products. An ontology consists of a set of concepts or terms and their relationships that describe some area of knowledge or build a representation of it (Swartout & Tate, 1999). Ontologies can be defined by identifying these concepts and the relationships between them and have simple rules to combine concepts for a particular domain. Representing products and product families by ontologies can provide solutions to promote component sharing, and assist designers search, explore, and analyze linguistic and parametric product family design information (Nanda et al., 2007).

MAIN FOCUS

This chapter describes a methodology for identifying a platform along with variant and unique modules in a product family using an ontology and data mining techniques. Figure 1 shows the proposed methodology that consists of three phases: (1) product representation, (2) module determination, and (3) platform and module identification. An ontology is used to represent products and components. Fuzzy clustering is employed to determine initial clusters based on the similarity among functional features. The clustering result is identified as the platform while modules are identified through the fuzzy set theory and classification. A description of each phase follows.

Figure 1. Methodology for platform and module identification



Phase 1: Product Representation

The basic idea of modular design is to organize products as a set of distinct components that can be designed independently and develop a variety of products through the combination and standardization of components (Kamrani & Salhieh, 2000). A product can be defined by its modules that consist of specific functions, and functions are achieved by the combination of the module attributes. To effectively define the relationship between functional hierarchies in a product, it is important to adopt an appropriate representation scheme for the products. The Techspecs Concept Ontology (TCO) is used to represent products and components (Moon et al., 2005). TCO provides functional representation-based semantics of products or components to better reflect customers' preferences and market needs.

Suppose that a product family consists of l products, $PF = (P_1, P_2, \ldots, P_l)$ and a product consists of m_i modules, $P_i = (\mathbf{x}_{i,1}, \mathbf{x}_{i,2},\ldots\mathbf{x}_{i,j},\ldots,\mathbf{x}_{i,m_i})$, where $\mathbf{x}_{i,j}$ is a module j in product i and consists of a vector of length n_m, $\mathbf{x}_{i,j} = (x_{i,j,1}, x_{i,j,2},\ldots x_{i,j,k},\ldots,x_{i,j,n_m})$, and the individual scalar components $x_{i,j,k}$ (k=1, 2,…, n_m) of a module $\mathbf{x}_{i,j}$ are called *functional features*. The functional feature consists of several attributes $a_{i,j,k,t}$ (t=1, 2,…, t_n), representing the function, $x_{i,j,k} = (a_{i,j,k,1}, a_{i,j,k,2},..., a_{i,j,k,t},..., a_{i,j,k,t_n})$, where t_n is the number of attributes represented by TCO.

Phase 2: Module Determination

Functional decomposition for a product can be represented in a hierarchical structure. A hierarchical clustering method can classify a set of objects by measuring the similarity between objects (Miyamoto, 1990). Because heuristic methods used to define a module may provide overlapping or non-crisp boundaries among module clusters (Stone et al., 2000), the results of traditional clustering approaches are not appropriate to define clusters as modules in product design. Fuzzy clustering approaches can use fuzziness related to product design features and provide more useful solutions (Liao, 2001). Fuzzy c-means clustering (FCM) (Bezdek, 1981) is employed to determine clusters for identifying modules for the product family. FCM is a clustering technique that is similar to k-means but uses fuzzy partitioning of data that is associated with different membership values between 0 and 1. Since FCM is an iterative algorithm, the aim in FCM is to find cluster centers that minimize a dissimilarity function.

Let X_k for k = 1, 2, …, n be a functional feature and a d-dimensional vector (d is the number of attributes), and u_{ik} the membership of X_k to the i-th cluster (i=1, 2, …, c). The u_{ik} representing a fuzzy case is between 0 and 1. For example, if $u_{ik} = 0$, u_{ik} has non-membership to cluster i, and if $u_{ik} = 1$, then it has full membership. Values in between 0 and 1 indicate fractional membership. Generally, FCM is defined as the solution of the following minimization problem (Bezdek, 1981):

$$J_{FCM}(U,V) = \{\sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ik})^m \left\|X_k - v_i\right\|^2\} \quad (1)$$

subject to:

$$\sum_{i=1}^{c} u_{ik} = 1, \text{ for all } k \quad (2)$$

$$u_{ik} \in [0,1] \quad (3)$$

where v_i is a cluster center of the i-th cluster that consists of a d-dimensional vector, and m is a parameter ($m \geq 1$) that plays a central role and indicates the fuzziness of clusters. An algorithm for solving this problem is introduced in Refs. (Bezdek, 1981; Torra, 2005). This FCM algorithm does not ensure that it converges to a global optimal solution; however, it always converges to a local optimum that may lead to a different local minima according to different initial cluster centers (Bezdek, 1981; Torra, 2005). The cluster number can be considered as the number of modules. A maximum membership value in clusters is an indicator for assigning to a module that can be considered as a group of similar functional features. Among clusters, clusters including the functional features for all selected products can be common modules for the platform.

In the proposed methodology, a coding approach is used to represent the attributes of components for a given clustering method. Each attribute takes a different code (number) based on functional features that are described as the functional basis proposed by Hirtz et al. (2002). The coding approach is problem-dependent, but an example can be found in the case study as it pertains to a power tool family.

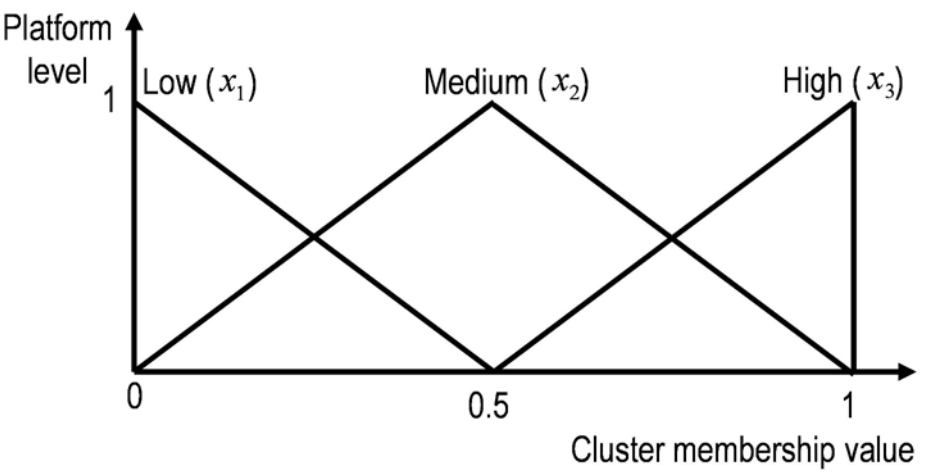
Phase 3: Platform and Module Identification

Classification is a learning technique to separate distinct classes and map data into one of several predefined classes (Johnson & Wichern, 2002). Classification approaches use classification rules that are usually developed from "learning" or "training" samples of pre-classified records (Braha, 2001; Johnson & Wichern, 2002). The process of generating classification rules is achieved using learning algorithms (Braha, 2001). Since the clusters in FCM are determined based on functional features, additional analysis is needed to define the modules. A designer can extract important design features that are represented by integration rules and split rules extracted from TCO. These design features are classified and translated into knowledge and rules for product design. Design rules can be translated into specific integration and split rules that can be used to classify clusters. If two clusters have the same module, then the modules are combined using the integration rule. Otherwise, if a cluster has several modules, then the cluster is divided into the number of products in the cluster using the split rule.

The clustering results provide membership values that represent the corresponding membership level of each cluster, which can be considered as the degree of similarity among functional features. Based on the fuzzy set theory (Zadeh, 1965), the membership values are measured in a rating scale of [0-1], and the ratings can be interpreted as fuzzy numbers based on different platform levels such as Low, Medium, and, High. Let X be a linguistic variable with the label "Platform level" with U = [0, 1]. Terms of this linguistic variable, fuzzy set, could be called "Low (x_1) - Medium (x_2) - High (x_3) platform level". As shown in Figure 2, the membership function of each fuzzy set is assumed to be triangular, and the platform level can take three different linguistic terms. Platform level membership functions are proposed to represent and determine the platform level of a common module. Therefore, the membership values of functions in a common module are transferred into platform level values by the platform level membership functions. The platform level of the common module is determined by the maximum value among average membership level values for the module.

The final results of the proposed methodology determine the platform along with the variant and unique modules for a product family. A platform consists of common modules with a high platform level that are shared. If variant modules are selected as a platform, additional functional features or costs will be required to make them a common module. Since the classification based on design rules considers the hierarchical relationship among functional features, modules can be designed independently. In the conceptual stages of design, these results can help decision-making by defining the set of modules for the product family. The effective set of modules will lead to improved product family design.

Figure 2. Fuzzy membership function representing platform level



CASE STUDY

To demonstrate the proposed methodology, a power tool family is investigated that consists of five distinct cordless power tools (see Table 1). Currently, these products only have common modules related to electrical components at the platform level. The proposed methodology determines if a more suitable platform and set of modules exists for the product family. This case study focuses on a function-based platform for the power tool family during the conceptual design phase. The products representation for the five tools was developed using TCO. Table 1 shows the 75 functional features of the selected five products. The five attributes of these functional features were coded using the values listed in the case study in Moon et al. (2006) as it pertains to a power tool family.

Fuzzy c-means clustering (FCM) was used to determine modules for the five products. Since the number of clusters affects the number of initial modules, it is important to select the number of clusters for FCM effectively. An optimal cluster number c ($6 \le c \le 74$) was estimated using the partition coefficient (PC) (Bezdek, 1981). For this case study, c = 13 was selected as the optimal cluster number to determine a platform and modules for the five products, since 12 to 15 clusters provided higher PC values than the other values. Table 2 shows the results of FCM using 13 clusters.

To determine a platform, 13 clusters were identified modules based on the integration rules and the split rules. For example, since two functional features ($x_{3,2,2}$, $x_{3,3,2}$) of the jig saw in Cluster 2 are related to different modules, a motor module and a conversion module, as shown in Table 2 the features of the jig saw were

Table 1. Product representation for the power tool family

Product	Module	Functional features
Circular saw 	$x_{1,1}$	1(3, 5, 5, 15, 1), 2(9, 5, 5, 15, 1), 3(5, 5, 5, 15, 1)
	$x_{1,2}$	1(13, 5, 5, 15, 1), 2(5, 9, 9, 2, 2)
	$x_{1,3}$	1(4, 0, 5, 15, 10), 2(5, 5, 5, 15, 10), 3(14, 5, 5, 15, 10)
	$x_{1,4}$	1(3, 1, 1, 2, 8), 2(14, 9, 9, 2, 8), 3(4, 9, 9, 2, 8)
	$x_{1,5}$	1(3, 1, 1, 2, 7), 2(14, 1, 1, 2, 7)
	$x_{1,6}$	1(3, 9, 9, 8, 2), 2(5, 9, 9, 8, 2)
Drill 	$x_{2,1}$	1(3, 5, 5, 15, 1), 2(9, 5, 5, 15, 1), 3(5, 5, 5, 15, 1)
	$x_{2,2}$	1(13, 5, 9, 2, 2), 2(5, 9, 9, 2, 2)
	$x_{2,3}$	1(4, 0, 5, 15, 10), 2(5, 5, 5, 15, 10), 3(14, 5, 5, 15, 10)
	$x_{2,4}$	1(3, 1, 1, 2, 4), 2(14, 9, 9, 2, 4), 3(4, 9, 9, 2, 4)
	$x_{2,5}$	1(3, 1, 1, 2, 7), 2(14, 1, 1, 2, 7)
	$x_{2,6}$	1(5, 9, 9, 8, 3), 2(11, 9, 9, 8, 3)
Jig saw 	$x_{3,1}$	1(3, 5, 5, 15, 1), 2(9, 5, 5, 15, 1), 3(5, 5, 5, 15, 1)
	$x_{3,2}$	1(13, 5, 9, 2, 2), 2(5, 9, 9, 2, 2)
	$x_{3,3}$	1(13, 9, 9, 8, 3), 2(5, 9, 9, 8, 3)
	$x_{3,4}$	1(4, 0, 5, 15, 10), 2(5, 5, 5, 15, 10), 3(14, 5, 5, 15, 10)
	$x_{3,5}$	1(3, 1, 1, 2, 7), 2(14, 1, 1, 2, 7)
	$x_{3,6}$	1(3, 1, 9, 2, 9), 2(14, 9, 9, 2, 9), 3(4, 9, 9, 2, 9)
Nailer 	$x_{4,1}$	1(3, 5, 5, 15, 1), 2(9, 5, 5, 15, 1), 3(5, 5, 5, 15, 1)
	$x_{4,2}$	1(13, 5, 9, 2, 2), 2(5, 9, 9, 2, 2)
	$x_{4,3}$	1(13, 9, 9, 3, 11), 2(5, 9, 9, 3, 11)
	$x_{4,4}$	1(4, 0, 5, 15, 10), 2(5, 5, 5, 15, 10), 3(14, 5, 5, 15, 10)
	$x_{4,5}$	1(3, 1, 1, 2, 7), 2(14, 1, 1, 2, 7)
	$x_{4,6}$	1(3, 1, 9, 2, 6), 2(4, 9, 9, 2, 6)
Sander 	$x_{5,1}$	1(3, 5, 5, 15, 1), 2(9, 5, 5, 15, 1), 3(5, 5, 5, 15, 1)
	$x_{5,2}$	1(13, 5, 9, 2, 2), 2(5, 9, 9, 2, 2)
	$x_{5,3}$	1(13, 9, 9, 22, 12), 2(5, 9, 9, 22, 12)
	$x_{5,4}$	1(4, 0, 5, 15, 10), 2(5, 5, 5, 15, 10), 3(14, 5, 5, 15, 10)
	$x_{5,5}$	1(3, 1, 1, 2, 7), 2(14, 1, 1, 2, 7)
	$x_{5,6}$	1(3, 1, 9, 2, 5), 2(14, 9, 9, 2, 5), 3(4, 9, 9, 2, 5)

Table 2. FCM results (membership value) for 13 clusters

Cluster	Circular saw	Drill	Jig saw	Nailer	Sander
1					$x_{5,3,1}(0.87)$, $x_{5,3,2}(0.79)$
2	$x_{1,2,2}(0.99)$, $x_{1,6,1}(0.33)$, $x_{1,6,2}(0.34)$	$x_{2,2,2}(0.99)$, $x_{2,4,3}(0.67)$, $x_{2,6,1}(0.3)$	$x_{3,2,2}(0.99)$, $x_{3,3,2}(0.3)$	$x_{4,2,2}(0.99)$	$x_{5,2,2}(0.99)$
3	$x_{1,4,3}(0.48)$		$x_{3,6,1}(0.23)$, $x_{3,6,3}(0.46)$	$x_{4,3,2}(0.38)$, $x_{4,6,1}(0.22)$, $x_{4,6,3}(0.46)$	$x_{5,6,1}(0.21)$, $x_{5,6,3}(0.36)$
4	$x_{1,3,3}(1)$	$x_{2,3,3}(1)$	$x_{3,4,3}(1)$	$x_{4,4,3}(1)$	$x_{5,4,3}(1)$
5	$x_{1,4,2}(0.71)$		$x_{3,6,2}(0.97)$	$x_{4,3,1}(0.92)$	
6				$x_{4,6,2}(0.74)$	
7	$x_{1,4,1}(0.99)$, $x_{1,5,1}(1)$	$x_{2,4,1}(0.89)$, $x_{2,5,1}(1)$	$x_{3,5,1}(1)$	$x_{4,5,1}(1)$	$x_{5,5,1}(1)$
8	$x_{1,3,1}(0.92)$, $x_{1,3,2}(0.89)$	$x_{2,3,1}(0.92)$, $x_{2,3,2}(0.89)$	$x_{3,4,1}(0.92)$, $x_{3,4,2}(0.89)$	$x_{4,4,1}(0.92)$, $x_{4,4,2}(0.89)$	$x_{5,4,1}(0.92)$, $x_{5,4,2}(0.89)$
9		$x_{2,4,2}(0.85)$, $x_{2,6,2}(0.24)$	$x_{3,3,1}(0.92)$		$x_{3,6,2}(0.57)$
10	$x_{1,1,1}(0.95)$, $x_{1,1,2}(0.77)$, $x_{1,1,3}(1)$	$x_{2,1,1}(0.95)$, $x_{2,1,2}(0.77)$, $x_{2,1,3}(1)$	$x_{3,1,1}(0.95)$, $x_{3,1,2}(0.77)$, $x_{3,1,3}(1)$	$x_{4,1,1}(0.95)$, $x_{4,1,2}(0.77)$, $x_{4,1,3}(1)$	$x_{5,1,1}(0.95)$, $x_{5,1,2}(0.77)$, $x_{5,1,3}(1)$
11	$x_{1,4,3}(0.48)$		$x_{3,6,1}(0.23)$, $x_{3,6,3}(0.46)$	$x_{4,3,2}(0.38)$, $x_{4,6,1}(0.22)$, $x_{4,6,3}(0.46)$	$x_{5,6,1}(0.21)$, $x_{5,6,3}(0.36)$
12	$x_{1,5,2}(1)$	$x_{2,5,2}(1)$	$x_{3,5,2}(1)$	$x_{4,5,2}(1)$	$x_{5,5,2}(1)$
13	$x_{1,2,1}(1)$	$x_{2,2,1}(1)$	$x_{3,2,1}(1)$	$x_{4,2,1}(1)$	$x_{5,2,1}(1)$

decomposed into two different modules using the split rules. Cluster 4 and Cluster 8 can be combined into one module using the integration rules, since these features in each product have been obtained from the same module and rules related to the functional features and the modules were generated. For instance, in the Cluster 4 and 8, the three functional features ($x_{1,3,1}$, $x_{1,3,2}$, $x_{1,3,3}$) of the circular saw have the integration rules.

Using the platform level membership function described in Phase 3, the platform levels of the clusters were determined. The current common modules in the actual product family shown in Table 3 are a battery module and a batter terminal connector in an input module. Based on the high platform level for the five power tools, a proposed platform consists of functions related to an electronic module, a motor module, a battery module, and an input module. Comparing this to the current platform for the five products, the number of common modules can be increased based on common functional features. This represents a 23% increase in the number of common modules within the family.

FUTURE TRENDS

In knowledge support and management systems, data mining approaches facilitate extraction of information in design repositories to generate new knowledge for product development. Knowledge-intensive and collaborative support has been increasingly important in

Table 3. Comparison to current products and proposed products

Current products		Proposed products	
Common	• Battery module ($x_{1,3,*}, x_{2,3,*}, x_{3,4,*}, x_{4,4,*}, x_{5,4,*}$) • Battery terminal connector ($x_{*,1,1}$)	Common	• Electronic module • Battery module • Motor module • Input module
Variant or Unique	• Electronic module ($x_{*,1,*}$) • Motor module ($x_{*,2,*}$) • Input module (x_{*5*}) • Gear train module ($x_{1,6,*}, x_{2,6,*}$) • Blade module ($x_{1,4,*}$, $x_{3,6,*}$) • Bit module ($x_{2,4,*}$) • Conversion module ($x_{3,3,*}, x_{5,3,*}$) • Nail hitter module ($x_{4,3,*}$) • Magazine module ($x_{4,6,*}$) • Sander M'G module ($x_{5,6,*}$)	Variant	• Gear train module • Conversion module
		Unique	• Blade module • Sander M'G module • Magazine module • Bit module ($x_{2,4,2}$) • Gear train module ($x_{2,6,2}$) • Conversion module ($x_{3,3,1}$) • Sander M'G module ($x_{5,6,2}$)

product development to maintain and create future competitive advantages (Zha & Sriram, 2006). A knowledge support system can provide a solution for iterative design and manufacturing activities that are performed by sharing and reusing knowledge related to product development processes. Representing products by ontologies will be an appropriate approach to support data mining techniques by capturing, configuring, and reasoning both linguistic and parametric design information in a knowledge management system effectively (Nanda et al., 2007).

CONCLUSION

Using an ontology and data mining techniques, a new methodology was described for identifying a module-based platform along with variant and unique modules in a product family. TCO was used to represent the functions of a product as functional hierarchies. Fuzzy c-means clustering was employed to cluster the functional features of products based on the similarity among them. The resulting clusters yield the initial modules, which are then identified through classification. Platform level membership functions were used to identify the platform within the family. The proposed methodology can provide designers with a module-based platform and modules that can be adapted to product design during conceptual design. Therefore, the methodology can help design a variety of products within a product family.

REFERENCES

Agard, B. & Kusiak, A. (2004). Data-mining-based methodology for the design of product family. *International Journal of Production Research, 42*(15), 2955-2969.

Bezdek, J. C. (1981). *Pattern recognition with fuzzy objective function algorithms*. New York, NY: Plenum.

Braha, D. (2001). *Data mining for design and manufacturing: Methods and applications.* Dordrecht, Netherlands: Kluwer Academic Publishers.

Hirtz, J., Stone, R. B., McAdams, D. A., Szykman, S. & Wood, K. L. (2002). A functional basis for engineering design: Reconciling and evolving previous efforts. *Research in Engineering Design, 13*(2), 65-82.

Jiao, J. & Zhang, Y. (2005). Product portfolio identification based on association rule mining. *Computer-Aided Design, 27*(2), 149-172.

Johnson, R. A. & Wichern, D. W. (2002). *Applied multivariate statistical analysis, Fifth Edition*. Upper Saddle River, NJ: Prentice Hall.

Kamrani, A. K. & Salhieh, S. M. (2000). *Product design for modularity.* Boston, MA.: Kluwer Academic Publishers.

Ma, M. Y., Chen, C. Y., & Wu, F. G. (2007). A design decison-making support model for customized product color combination. *Computer in Industry, 58*(6), 504-518.

Miyamoto, S. (1990). *Fuzzy sets in information retrieval and cluster analysis.* Dordrecht, Netherlands: Kluwer Academic Publishers.

Moon, S. K., Kumara, S. R. T. & Simpson, T. W. (2005). Knowledge representation for product design using Techspecs Concept Ontology. *The IEEE International Conference on Information Reuse and Integration* (pp. 241-246), Las Vegas, NV.

Moon, S. K., Kumara, S. R. T. & Simpson, T. W. (2006). Data mining and fuzzy clustering to support product family design. *ASME Design Engineering Technical Conferences - Design Automation Conference* (Paper No. DETC2006/DAC-99287), Philadelphia, PA: ASME.

Nanda, J., Thevenot, H. J., Simpson, T. W., Stone, R. B., Bohm, M. & Shooter, S. B. (2007). Product family design knowledge representation, aggregation, reuse, and analysis. *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing, 21*(2), 173-192.

Shooter, S. B., Simpson, T. W., Kumara, S. R. T., Stone, R. B. & Terpenny, J. P. (2005). Toward an information management infrastructure for product family planning and platform customization. *International Journal of Mass Customization, 1*(1), 134-155.

Simpson, T. W. (2004). Product platform design and customization: Status and promise. *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing, 18*(1), 3-20.

Simpson, T. W., Siddique, Z., & Jiao, J. (2005). *Product platform and product family design: Methods and applications*. New York, YN: Springer.

Stone, R. B., Wood, K. L. & Crawford, R. H. (2000). A heuristic method for identifying modules for product architectures. *Design Studies, 21*(1), 5-31.

Swartout, W. & Tate, A. (1999). Ontologies. *IEEE Transactions on Intelligent Systems, 14*(1), 18-19.

Torra, V. (2005). Fuzzy c-means for fuzzy hierarchical clustering. *The IEEE International Conference on Fuzzy Systems* (pp. 646-651), Reno, NV.

Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, 8, 338-353.

Zha, X. F. & Sriram, R. D. (2006). Platform-based product design and development: A knowledge-intensive support approach. *Knowledge-Based Systems, 19*(7), 524-543.

KEY TERMS

Classification: A learning technique to separate distinct classes and map data into one of several predefined classes.

Design Knowledge: The collection of knowledge that can support the design activities and decision-making in product development, and can be represented as constraints, functions, rules, and facts that are associated with product design information.

Fuzzy C-Mean (FCM) Clustering: A clustering technique that is similar to k-means but uses fuzzy partitioning of data that is associated with different membership values between 0 and 1.

Modular Design: A design method to organize products as a set of distinct components that can be designed independently and develop a variety of prod-

ucts through the combination and standardization of components.

Ontology: A set of concepts or terms and their relationships that describe some area of knowledge or build a representation of it.

Product Family: A group of related products based on a product platform, facilitating mass customization by providing a variety of products for different market segments cost-effectively.

Product Platform: A set of features, components or subsystems that remain constraint from product to product, within a given product family.

Data Mining on XML Data

Qin Ding
East Carolina University, USA

INTRODUCTION

With the growing usage of XML data for data storage and exchange, there is an imminent need to develop efficient algorithms to perform data mining on semi-structured XML data. Mining on XML data is much more difficult than mining on relational data because of the complexity of structure in XML data. A naïve approach to mining on XML data is to first convert XML data into relational format. However the structure information may be lost during the conversion. It is desired to develop efficient and effective data mining algorithms that can be directly applied on XML data.

BACKGROUND

In recent years, XML has become very popular for representing semi-structured data and a standard for data exchange over the web. XML stands for Extensible Markup Language. It is a simple and very flexible text format derived from SGML (Standard Generalized Markup Language). Both XML and SGML are meta-languages because they are used for defining markup language. Originally designed to meet the challenges of large-scale electronic publishing, XML is also playing an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere.

Below is a simplified example of an XML document. As can be seen, elements (or called tags) are the primary building blocks of an XML document. Unlike HTML which uses a fixed set of tags, XML allows user to define new collections of tags that can be used to structure any type of data or document. An element can have descriptive attributes that provide additional information about the element. Document Type Definitions (DTDs) can be used to specify which elements and attributes we can use and the constraints on these elements, such as how elements can be nested. XML allows the representation of semi-structured and hierarchal data containing not only the values of individual items but also the relationships between data items.

Data mining is the process to extract useful patterns or knowledge from large amount of data. As the amount of available XML data is growing continuously, it will be interesting to perform data mining

Figure 1. An XML document

```
<Department>
   <People>
     <Employee>
       <PersonalInfo> …</PersonalInfo>
       <Education> … </Education>
       <Publications>
           <Book year=”2002” name=”XML Query Languages”>
              <Author> … </Author>
              <Publisher> … </Publisher>
              <Keyword>XML</Keyword> … <Keyword>XQuery</Keyword>
          </Book>
          <Journal year=“2000” vol=”4” name=”DMKD” Publisher=”Kluwer”>
              <Author> ... </Author>>
              <Keyword>RDF</Keyword> … <Keyword>XML</Keyword>
          </Journal>
       </Publications>
     </Employee>
   </People>
</Department>
```

on XML data so that useful patterns can be extracted from XML data. From the example in Figure 1, we might be able to discover such patterns as "researchers who published about XML also published something related to XQuery" where "XML" and "XQuery" are keywords of the publications. This interesting pattern can be represented as an association rule in the format of "XML => XQuery". The task of data mining on XML data can be significant, yet challenging, due to the complexity of the structure in XML data.

Data mining has been successfully applied to many areas, ranging from business, engineering, to bioinformatics (Han & Kamber, 2006). However, most data mining techniques were developed for data in relational format. In order to apply these techniques to XML data, normally we need to first convert XML data into relational data format, and then traditional data mining algorithms can be applied on converted data. One way to map XML data into relational schema is to decompose XML documents entirely, remove the XML tags, and store the element and attribute values in relational tables. In this process, most aspects of the XML document structure are usually lost, such as the relative order of elements in the document, and the nested structure of the elements.

In order to avoid mapping the XML data to relational format, researchers have been trying to utilize XQuery, the W3C (World Wide Web Consortium) standard query language for XML, to support XML mining. On the one hand, XQuery provides a flexible way to extract XML data; on the other hand, by adding another layer using XQuery, the mining efficiency may be greatly affected. In addition, a query language such as XQuery may have limited querying capabilities to support all the data mining functionalities.

MAIN FOCUS

Data mining on XML data can be performed on the content as well as the structure of XML documents. Various data mining techniques can be applied on XML data, such as association rule mining and classification. In this section, we will discuss how to adapt association rule mining and classification techniques to XML data, for example, how to discover frequent patterns from the contents of native XML data and how to mine frequent tree patterns from XML data. We will also discuss other work related to XML data mining, such as mining XML query patterns (Yang et al, 2003) and using XML as a unified framework to store raw data and discovered patterns (Meo & Psaila, 2002).

Association Rule Mining on XML Data

Association rule mining is one of the important problems in data mining (Agrawal et al, 1993; Agrawal & Srikang, 1994; Han et al, 2000). Association rule mining is the process of finding interesting implication or correlation within a data set. The problem of association rule mining typically includes two steps. The first step is to find frequently occurring patterns. This step is also called "Frequent Pattern Mining". The second step is to discover interesting rules based on the frequent patterns found in the previous step. Most association rule algorithms, such as Apriori (Agrawal & Srikant, 1994) and FP-growth (Han et al, 2000), can only deal with flat relational data.

Braga et al addressed the problem of mining association rules from XML data in their recent work (Braga et al, 2002; Braga et al, 2003). They proposed an operator called "XMINE" to extract association rules from native XML documents. The XMINE operator was inspired by the syntax of XQuery and was based on XPath, which is the XML path language, an expression language for navigating XML document. Using the example in Figure 1, the "XMINE" operator can be used, as shown in Figure 2, to discover rules such as "XML => XQuery", where both sides of the rule have the same path, which is specified as "ROOT/Publications//Keyword" using the XPath language. The limitation of this work is that it only focuses on mining specific rules with pre-specified antecedent (the left-hand side) and consequence (the right-hand side), while the general association rule mining should target all the possible rules among any data items (Ding et al, 2003).

Dobbie and Wan proposed an XQuery-based Aprioir-like approach to mining association rules from XML data (Dobbie & Wan, 2003). They show that any XML document can be mined for association rules using only XQuery without any pre-processing and post-processing. However, since XQuery is designed only to be a general-purpose query language, it puts a lot of restriction on using it to do complicated data mining process. In addition, the efficiency is low due to the extra cost of XQuery processing.

Another algorithm called TreeFinder was introduced by Termier et al (Termier et al, 2002). It aims at search-

Figure 2. Operations to mine rules such as "XML => XQuery"

```
In document("http://www.cs.atlantis.edu/staff.xml")
XMINE RULE
FOR ROOT IN /Department/Employee
LET BODY := ROOT/Publications//Keyword,
   HEAD := ROOT/Publications//Keyword
EXTRACTING RULES WITH SUPPORT = 0.1 AND CONFIDENCE = 0.4
```

ing frequent tress from a collection of tree-structured XML data. It extends the concept of "frequent itemset" to "frequent tree structure". One limitation of this algorithm is that, as mentioned in the paper, TreeFinder is correct but not complete; it only finds a subset of the actually frequent trees.

Ding and Sundarraj proposed an approach to XML association rule mining without using XQuery (Ding & Sundarraj, 2006). Their approach adapted the Apriori and FP-growth algorithms to be able to support association rule mining on XML data. Experimental results show that their approach achieves better performance than XQuery-based approaches.

Data preprocessing on XML documents might be needed before performing the data mining process. XSLT (Extensible Stylesheet Language Transformation) can be used to convert the content of an XML document into another XML document with different format or structure.

The open issue in XML association rule mining is how to deal with the complexity of the structure of XML data. Since the structure of the XML data can be very complex and irregular, identifying the mining context on such XML data may become very difficult (Wan & Dobbie, 2003).

Classification on XML Data

Classification is another important problem in data mining (Breiman et al, 1984; Quinlan, 1993). Similar to association rule mining, many classification algorithms only deal with relational data. The problem of classification on XML data has not been well studied. Current classification methods for XML documents use information retrieval based methods. These methods simply treat an XML document as a regular text document with a bag of words, so XML mining becomes the regular text mining; as a result, a significant amount of structural information hidden in the documents was ignored. In many cases, the classification information is actually hidden in the structural information.

Zaki and Aggarwal proposed an effective rule-based classifier called XRules for XML data (Zaki & Aggarwal, 2003). XRules mines frequent structures from the XML documents in order to create the structural classification rules. Their results show that structural classifier outperforms IR-based classifiers. This work is related to Termier's TreeFinder algorithm (Termier et al, 2002) as they both focus on structure mining on XML data.

Mining Frequent XML Query Patterns

Data mining techniques can also be applied to discover frequent query patterns. Response time to XML query has always been an important issue. Most approaches concentrate on indexing XML documents and processing regular path expression. Yang et al indicated that discovering frequent patterns can improve XML query response time since the answers to these queries can be stored and indexed (Yang et al, 2003). They proposed a schema-guided mining approach to discover frequent rooted subtrees from XML queries. The basic idea is that an XML query can be transformed into a query pattern tree. By computing the frequency counts of rooted subtrees, frequent query patterns can be discovered.

Building XML-Based Knowledge Discovery Systems

XML has also become a format to represent the discovered patterns from data mining. For example, it has become a convention to output the association rules, derived from the input XML data, using XML representation.

Meo & Psaila proposed a data model that uses XML as the unified framework to describe source raw data, heterogeneous mined patterns and data mining state-

ments, so that they can be stored inside a unique XML-based inductive database (Meo & Psaila, 2002).

FUTURE TRENDS

XML data mining research is still in its infant stage. However it will become increasingly important as the volume of XML data grows rapidly and the XML-related technology enhances greatly. So far, only a limited number of areas in XML mining have been explored, such as association rule mining and classification on XML. We believe that there are many other open problems in XML mining that need to be tackled, such as clustering XML documents, discovering sequential patterns in XML data, XML outlier detection, privacy-preserving data mining on XML, to name just a few. How to integrate content mining and structure mining on XML data is also an important issue.

As more researchers become interested and involved in XML data mining research, it is possible and beneficial to develop benchmarks on XML data mining. New XML data mining applications may also emerge, for example, XML mining for Bioinformatics.

CONCLUSION

Since XML is becoming a standard to represent semi-structured data and the amount of XML data is increasing continuously, it is of importance yet challenging to develop data mining methods and algorithms to analyze XML data. Mining of XML data significantly differs from structured data mining and text mining. XML data mining includes both content mining and structure mining. Most existing approaches either utilize XQuery, the XML query language, to support data mining, or convert XML data into relational format and then apply the traditional approaches. It is desired that more efficient and effective approaches be developed to perform various data mining tasks on semi-structured XML data.

REFERENCES

Agrawal, R., Imielinski, T., & Swami, A. (1993). Mining Association Rules Between Sets of Items in Large Database. *Proceedings of ACM SIGMOD International Conference on Management of Data*, 207-216.

Agrawal, R., & Srikant, R. (1994). Fast Algorithms for Mining Association Rules. *Proceedings of International Conference on Very Large Data Bases,* 487-499.

Braga,D., Campi,A., Ceri, S., Klemettinen, M., & Lanzi, PL. (2003). Discovering Interesting Information in XML Data with Association Rules. *Proceedings of ACM Symposium on Applied Computing*, 450-454.

Braga, D., Campi, A., Ceri, S., Klemettinen, M., & Lanzi, PL. (2002). Mining Association Rules from XML Data. *Proceedings of International Conference on Database and Expert System Applications*, Lecture Notes in Computer Science, Vol. 2454, 21-30.

Braga, D., Campi, A., Ceri, S., Klemettinen, M. & Lanzi, PL. (2002). A Tool for Extracting XML Association Rules from XML Documents. *Proceedings of IEEE International Conference on Tools with Artificial Intelligence*, 57-64.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and Regression Trees*, Wadsworth, Belmont.

Ding, Q., Ricords, K., & Lumpkin, J. (2003). Deriving General Association Rules from XML Data. *Proceedings of International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, 348-352.

Ding, Q., & Sundarraj, G, (2006). Association Rule Mining from XML Data. *Proceedings of International Conference on Data Mining*, 144-150.

Han, J., & Kamber, M. (2006). *Data Mining: Concepts and Techniques*. Morgan Kaufmann.

Han, J., Pei, J., & Yin, Y. (2000). Mining Frequent Patterns without Candidate Generation. *Proceedings of ACM SIGMOD International Conference on Management of Data*, 1-12.

Meo, R., & Psaila, G. (2002). Toward XML-based Knowledge Discovery Systems. *Proceedings of IEEE International Conference on Data Mining*, 665-668.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann.

Termier, A., Rousset, M-C., & Sebag, M. (2002). TreeFinder: A First Step towards XML Data Mining. *Proceedings of IEEE International Conference on Data Mining*, 450-457.

Wan, J. W. W., & Dobbie, G. (2003). Extracting association rules from XML documents using XQuery. *Proceedings of the ACM International Workshop on Web Information and Data Management*, 94–97.

Yang, L. H., Lee, M. L., Hsu, W., & Acharya, S. (2003). Mining Frequent Query Patterns from XML Queries. *Proceedings of the International Conference on Database Systems for Advanced Applications*, 355-362.

Zaki, M. J., & Aggarwal, C. (2003). XRULES: An Effective Structural Classifier for XML Data. *Proceedings of International Conference on Knowledge Discovery and Data Mining*, 316-325.

KEY TERMS

Association Rule Mining: The process of finding interesting implication or correlation within a data set.

Data Mining: The process of extracting non-trivial, implicit, previously unknown and potentially useful patterns or knowledge from large amount of data.

Document Type Definition (DTD): A DTD is the formal definition of the elements, structures, and rules for marking up a given type of SGML or XML document.

Semi-Structured Data: The type of data with implicit and irregular structure but no fixed schema.

SGML (Standard Generalized Markup Language): SGML is a meta-language for how to specify a document markup language or tag sets.

XML (Extensible Markup Language): XML is a W3C-recommended general-purpose markup language for creating special-purpose markup languages. It is a simplified subset of SGML, capable of describing many different kinds of data.

XPath (XML Path Language): XPath is a language for navigating in XML documents. It can be used to address portions of an XML document.

XQuery: XQuery is the W3C standard query language for XML data.

XSLT (Extensible Stylesheet Language Transformation): XSLT is a language for transforming XML documents into other XML documents with different format or structure.

W3C (World Wide Web Consortium): An international consortium that develops interoperable technologies (specifications, guidelines, software, and tools) related to the internet and the web.

Data Mining Tool Selection

Christophe Giraud-Carrier
Brigham Young University, USA

INTRODUCTION

It is sometimes argued that all one needs to engage in Data Mining (DM) is data and a willingness to "give it a try." Although this view is attractive from the perspective of enthusiastic DM consultants who wish to expand the use of the technology, it can only serve the purposes of one-shot proofs of concept or preliminary studies. It is not representative of the complex reality of deploying DM within existing business processes. In such contexts, one needs two additional ingredients: a process model or methodology, and supporting tools.

Several Data Mining process models have been developed (Fayyad et al, 1996; Brachman & Anand, 1996; Mannila, 1997; Chapman et al, 2000), and although each sheds a slightly different light on the process, their basic tenets and overall structure are essentially the same (Gaul & Saeuberlich, 1999). A recent survey suggests that virtually all practitioners follow some kind of process model when applying DM and that the most widely used methodology is CRISP-DM (KDnuggets Poll, 2002). Here, we focus on the second ingredient, namely, supporting tools.

The past few years have seen a proliferation of DM software packages. Whilst this makes DM technology more readily available to non-expert end-users, it also creates a critical decision point in the overall business decision-making process. When considering the application of Data Mining, business users now face the challenge of selecting, from the available plethora of DM software packages, a tool adequate to their needs and expectations. In order to be informed, such a selection requires a standard basis from which to compare and contrast alternatives along relevant, business-focused dimensions, as well as the location of candidate tools within the space outlined by these dimensions. To meet this business requirement, a standard schema for the characterization of Data Mining software tools needs to be designed.

BACKGROUND

The following is a brief overview, in chronological order, of some of the most relevant work on DM tool characterization and evaluation.

Information Discovery, Inc. published, in 1997, a taxonomy of data mining techniques with a short list of products for each category (Parsaye, 1997). The focus was restricted to implemented DM algorithms.

Elder Research, Inc. produced, in 1998, two lists of commercial desktop DM products (one containing 17 products and the other only 14), defined along a few, yet very detailed, dimensions (Elder & Abbott, 1998; King & Elder, 1998). Another 1998 study contains an overview of 16 products, evaluated against pre-processing, data mining and post-processing features, as well as additional features such as price, platform, release date, etc. (Gaul & Saeuberlich, 1999). The originality of this study is its very interesting application of multi-dimensional scaling and cluster analysis to position 12 of the 16 evaluated tools in a four-segment space.

In 1999, the Data & Analysis Center for Software (DACS) released one of its state-of-the-art reports, consisting of a thorough survey of data mining techniques, with emphasis on applications to software engineering, which includes a list of 55 products with both summary information along a number of technical as well as process-dependent features and detailed descriptions of each product (Mendonca & Sunderhaft, 1999). Exclusive Ore, Inc. released another study in 2000, including a list of 21 products, defined mostly by the algorithms they implement together with a few additional technical dimensions (Exclusive Ore, 2000).

In 2004, an insightful article discussing high-level considerations in the choice of a DM suite—highlighting the fact that no single suite is best overall—together with a comparison of five of the then most widely used commercial DM software packages, was published in a well-read trade magazine (Nisbett, 2004). About the same time, an extensive and somewhat formal DM tool

characterization was proposed, along with a dynamic database of the most popular commercial and freeware DM tools on the market (Giraud-Carrier & Povel, 2003).[1] In a most recent survey of the field, the primary factors considered in selecting a tool were highlighted, along with a report on tool usage and challenges faced (Rexer et al, 2007).

Finally, it is worth mentioning a number of lists of DM tools that, although not including any characterization or evaluation, provide an useful starting point for tool evaluation and selection exercises by centralizing (and generally maintaining in time) basic information for each tool and links to the vendor's homepage for further details (KDNet, 2007; KDnuggets, 2007; Togaware, 2007).

MAIN FOCUS

The target audience of this chapter is business decision-makers. The proposed characterization, and accompanying database, emphasize the complete Data Mining process and are intended to provide the basis for informed, business-driven tools comparison and selection. Much of the content of this chapter is an extension of the work in (Giraud-Carrier & Povel, 2003), with additional insight from (Worthen, 2005; Smalltree, 2007; SPSS, 2007).

The set of characteristics for the description of DM tools can be organized naturally in a hierarchical fashion, with the top level as depicted in Figure 1. Each branch is expanded and discussed in the following sections.

Business Context and Goals

Data mining is primarily a *business-driven process* aimed at the discovery and consistent use of profitable knowledge from corporate data. Naturally, the first questions to ask, when considering the acquisition of supporting DM tools, have to do with the business context and goals, as illustrated in Figure 2.

Since different business contexts and objectives call for potentially different DM approaches, it is critical to understand what types of questions or business problems one intends to solve with data mining, who will be responsible for executing the process and presenting the results, where the data resides, and how the results of the analysis will be disseminated and deployed to the business. Answers to these high-level questions provide necessary constraints for a more thorough analysis of DM tools.

Model Types

The generative aspect of data mining consists of the building of a model from data. There are many available (machine learning) algorithms, each inducing one of a variety of types of models, including predictive models (e.g., classification, regression), descriptive models (e.g., clustering, segmentation), dependency

Figure 1. Top level of the DM tool description hierarchy

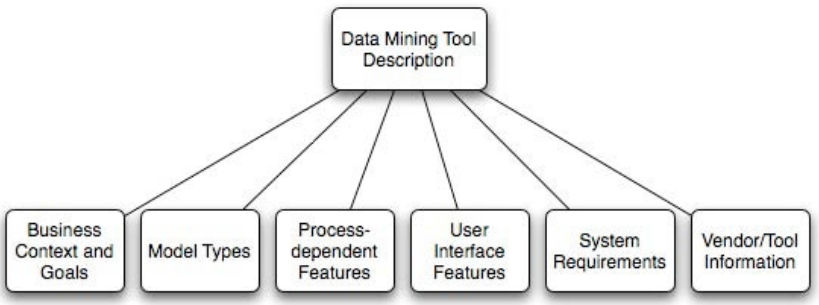
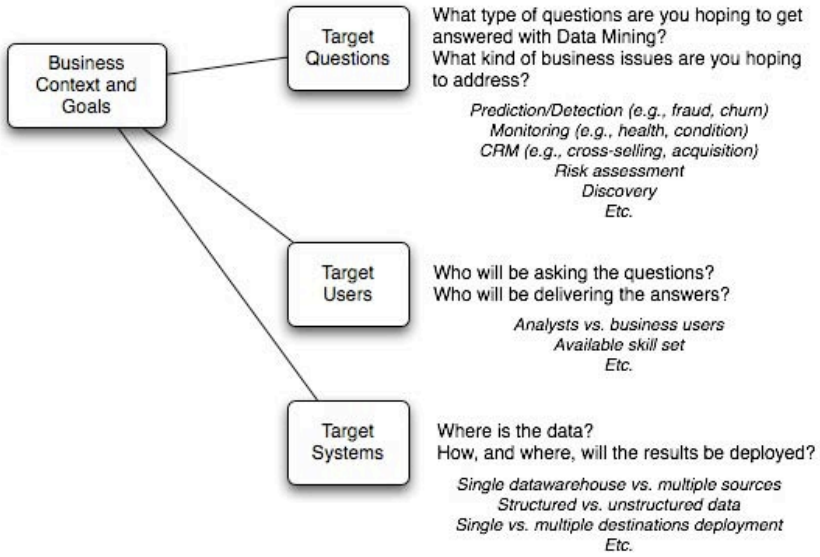


Figure 2. Business context and goals description



models (e.g., association), anomaly or novelty detection models, and time-series analysis models.

Given the dependency between business objectives and classes of models, one must ensure that the candidate DM tools support the kinds of modeling algorithms necessary to address the business questions likely to arise.

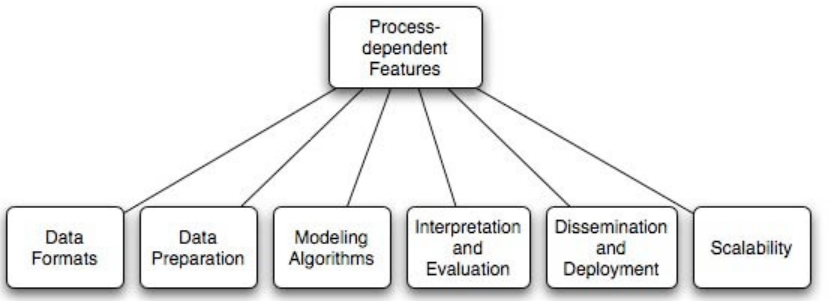
Process-Dependent Features

The data mining process is iterative by nature, and involves a number of complementary activities (e.g., CRISP-DM has 6 phases). Just as business environments differ, DM tools differ in their functionality and level of support of the DM process. Some are more complete and versatile than others. Users must therefore ensure that the selected tool supports their needs, standards and practices.

The main issues associated with process-dependent features and relevant to DM tool selection are summarized in Figure 3. A short description and discussion of each feature follows.

1. Data Formats: Data generally originates in a number of heterogeneous databases in different formats. To avoid unnecessary "interfacing" work, users should strive to select a tool able to read their various data formats. Most commonly DM tool supported formats/connectivity include flat file (e.g., comma-separated), ODBC/JDBC, SAS, and XML.
2. Data Preparation: Raw data is seldom readily usable for DM. Instead, before it can be further analyzed, it may be usefully enriched and transformed, e.g., to add new features or to remove noise. Users must not underestimate the need for pre-processing and data preparation. Experience suggests that such activities represent the bulk of the DM process, reaching as much as 80% of the overall process' time. It is thus critical to ensure that the tool selected is able to assist effectively, both from an usability point of view and from a functionality perspective. Data preparation, or pre-processing techniques, can be broadly categorized as follows.

Figure 3. Process-dependent features description



- Data Characterization: statistical and information theoretic measurements such as mean value, mutual entropy, kurtosis, etc.
- Data Visualization: graphical representations to quickly "eye-ball" underlying distributions, data spread, etc.
- Data Cleaning: automatic detection of undesirable conditions, such as outliers, single-valued attributes, identifier attributes (i.e., key attributes with as many values as there are records), records or attributes with too many missing values, etc.
- Record Selection: mechanisms to select specific records (or rows) based on ad-hoc queries.
- Attribute Selection: mechanisms to select specific attributes (or columns) based on ad-hoc queries.
- Data Transformation: mechanisms to transform data such as discretization, definition of new attributes, thresholding, etc.

3. Modeling Algorithms: There are a number of algorithmic techniques available for each DM approach, with features that must be weighed against data characteristics and additional business requirements (e.g., understandability). Users must ensure that their chosen tool implements algorithms that meet their needs. The following are the standard classes of DM modeling algorithms.
   - Decision Trees
   - Rule Learning
   - Neural Networks
   - Linear/Logistic Regression
   - Kernel-based Learning (e.g., SVM)
   - Association Learning
   - Instance-based/Nearest-neighbor Learning
   - Unsupervised Learning
   - Probabilistic Learning
   - Reinforcement Learning
   - Sequence Learning
   - Text Mining

   Note that there is a natural dependency between modeling algorithm classes and model types, as each modeling algorithm class is targeted at a specific type of model (Giraud-Carrier & Povel, 2003). If the potential uses of DM are varied, one may choose a tool that implements a great variety of techniques. On the other hand, if the application is fairly narrow and well-defined, one may opt for a specialized tool implementing a technique proven to be most suitable to the target application.
4. Evaluation and Interpretation: DM will produce results. These, however, are not always actionable or profitable. Ultimately, the "buck stops with the business users." Hence, users must be able to "see" enough of what they need in order to assess the results' potential value to the business. The

following are the standard methods of DM results evaluation. These are not mutually exclusive, but rather complementary in many cases.
   - Hold-out/Independent Test Set
   - Cross-validation
   - Lift/Gain Charts
   - ROC Analysis
   - Summary Reports (e.g., confusion matrix, F-measure)
   - Model Visualization
5. Dissemination and Deployment: In order to "close the loop," knowledge acquired through Data Mining must become part of business-as-usual. Hence, users must decide on what and how they wish to disseminate/deploy results, and how they integrate DM into their overall business strategy. The selected tool, or custom implementation, must support this view. Available methods are as follows.
   - Comment Fields
   - Save/Reload Models
   - Produce Executable
   - PMML/XML Export
6. Scalability: It is not unusual to be dealing with huge amounts of data (e.g., bank transactions) and/or to require near real-time responses (e.g., equipment monitoring). In such cases, users must consider how well the selected tool scales with the size of their data sets and their time constraints. Three features capture this issue, namely, dataset size limit, support for parallelization, and incrementality (of the modeling algorithms).

In addition to the "core" elements described above, several tools offer a number of special features that facilitate or enhance the work of users. The following are particularly relevant:

1. Expert Options: Most algorithms require the setting of numerous parameters, which may have significant impact on their performance. Typical DM tools implement the vanilla version of these algorithms, with default parameter settings. Although useful to novice users, this is limiting for advanced users who may be able to leverage the algorithms' parametrizability.
2. Batch Processing: For applications requiring the generation of many models or the re-creation of new models (e.g., customer targeting in successive marketing campaigns), the ability to run algorithms in batch mode greatly increases users effectiveness.

User Interface Features

Most of the algorithms (both for pre-processing and modeling) used in Data Mining originate from research labs. Only in the past few years have they been incorporated into commercial packages. Users differ in skills (an issue raised above under Business Context and Goals), and DM tools vary greatly in their type and style of user interaction. One must think of the DM tool target audience and be sure to select a tool that is adapted to the skill level of such users. In this dimension, one may focus on three characteristics:

- Graphical Layout: Does the selected tool offer a modern GUI or is it command-line driven? Are the DM process steps presented in an easy-to-follow manner?
- Drag&Drop or Visual Programming: Does the selected tool support simple visual programming based on selecting icons from palettes and sequencing them on the screen?
- On-line Help: Does the tool include any on-line help? How much of it and of what quality for the inexperienced user?

System Requirements

Software tools execute in specific computer environments. It is therefore important for users to be aware of the specific requirements of their selected tool, such as hardware platform (e.g., PC, Unix/Solaris workstation, Mac, etc.), additional software requirements (DB2, SAS Base, Oracle, Java/JRE, etc.), and software architecture (e.g., standalone vs. client/server).

One should carefully consider what the impact of deploying the tool would have on the existing IT infrastructure of the business. Will it require additional investments? How much work will be needed to interface the new software with existing one?

Vendor/Tool Information

In addition to, or sometimes in spite of, the technical aspects described above, there are also important com-

mercial issues, which must be addressed when selecting a software package. These may for example include:

- Availability of several contact points: 24-hr toll-free phone number, Web site's FAQ, email address for general inquiries, etc.
- Level of technical support: How well is the vendor/distributor able to carry out such activities as maintenance, upgrades, technical support, help desk, etc.?
- Price: How much does the tool cost (for 1 license on 1 machine)? What licensing options exist?
- Availability of a free demo or evaluation version: Is it possible to download (or otherwise obtain) a free evaluation copy of the software?
- Training needed: Does our current skill set support the use of the tool? If not, how much training (time and effort) will my users need before they can successfully leverage the DM tool's strength?
- Company's market penetration: How much of the DM tool market does the vendor hold (i.e., number of customers/sites)? What types of customers (e.g., activity sector, size, etc.)?
- Successful case studies: What is the tool's track record? Are there documented cases of successful applications of the tool to a variety of business problems, including the ones we are interested in?
- Company's longevity: How long has the vendor been in business? How likely is it to continue?

FUTURE TRENDS

The DM tool characterization presented here offers business users a standard way to compare and contrast candidate DM tools against each other, as well as against their own business requirements. However, the multi-dimensional nature of the characterizations, together with the many possible business constraints and preferences, mean that a direct comparison may be unwieldy. However, the characterizations may be used as inputs into semi-automatic methodologies that assist users in evaluating and selecting an appropriate tool, based on a weighted score of a large number of relevant criteria (Collier et al, 1999; Britos et al, 2006).

CONCLUSION

A general schema for the characterization of Data Mining tools, organized around the several phases of the DM process, has been outlined. The dimensions proposed are meant to be accessible to business users and serve to assist them as a first, essential step in the DM tool selection process.

REFERENCES

Brachman, R.J., & Anand, T. (1996). The Process of Knowledge Discovery in Databases: A Human-centered Approach, in Fayyad, U.M., Piatetsky-Shapiro, G., & Smyth, P. (Eds.), *Advances in Knowledge Discovery and Data Mining*, AAAI/MIT Press, 33-51.

Britos, P., Merlino, H., Fernández, E., Ochoa, M., Diez, E. & García-Martínez, R. (2006). Tool Selection Methodology in Data Mining, in *Proceedings of the Fifth Ibero-American Symposium on Software Engineering*, 85-90.

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). *CRISP-DM 1.0: Step-by-step Data Mining Guide*, SPSS, Inc.

Collier, K., Carey, B., Sautter, D., & Marjaniemi, C. (1999). A Methodology for Evaluating and Selecting Data Mining Software, in *Proceedings of the Thirty Second Hawaii International Conference on System Sciences*, available online at http://www.cse.nau.edu.

Elder, J.F., & Abbott, D.W. (1998). A Comparison of Leading Data Mining Tools, Tutorial presented at *the Fourth Annual Conference on Knowledge Discovery and Data Mining*, available online at http://www.dataminginglab.com.

Exclusive Ore, Inc. (2000). Data Mining Product Features, available online at http://www.xore.com/prodtable.html.

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). The KDD Process for Extracting Useful Knowledge from Volumes of Data, *Communications of the ACM*, 39(11):27-34.

Gaul, W., & Saeuberlich, F. (1999). Classification and Positioning of Data Mining Tools, in Gaul, W., & Lo-

carek-Junge, H. (Eds.), *Classification in the Information Age*, Springer, Berlin, Heidelberg, 143-152.

Giraud-Carrier, C., & Povel, O. (2003). Characterising Data Mining Software, *Journal of Intelligent Data Analysis*, **7**(3):181-192.

Goebel, M., & Gruenwald, L. (1999). A Survey of Data Mining and Knowledge Discovery Software Tools, *SIGKDD Explorations*, **1**(1):20-33.

KDNet (2007). Software, available online at http://www.kdnet.org/kdnet/control/software.

KDnuggets (2007). Software for Data Mining and Knowledge Discovery, available online at http://kdnuggets.com/software/index.html.

KDnuggets Poll (2002). What Main Methodology Are You Using for Data Mining? available online at http://www.kdnuggets.com/polls/methodology.htm.

King, M.A., & Elder, J.F. (1998). Evaluation of Fourteen Desktop Data Mining Tools, in *Proceedings of the 1998 IEEE International Conference on Systems, Man and Cybernetics*, available online at http://www.datamininglab.com.

Mannila, H. (1997). Methods and Problems in Data Mining, in *Proceedings of the 1997 International Conference on Database Theory.*

Mendonca, M., & Sunderhaft, N.L. (1999). Mining Software Engineering Data: A Survey, State-of-the-Art Report DACS-SOAR-99-3, Data & Analysis Center for Software (DACS). available online at http://www.dacs.dtic.mil/techs/datamining/index.html.

Nisbett, R.A. (2004). How to Choose a Data Mining Suite, *DM Direct Special Report*, 23 March, available online from http://www.dmreview.com/editorial.

Parsaye, K. (1997). A Characterization of Data Mining Technologies and Processes, *Journal of Data Warehousing*, December Issue, available online at http://www.datamining.com/dm-tech.htm.

Rexer, K., Gearan, P., & Allen, H.N. (2007). *Surveying the Field: Current Data Mining Applications, Analytic Tools, and Practical Challenges*, Rexer Analytics, available from krexer@RexerAnalytics.com.

Smalltree, H. (2007). Gartner Customer Data Mining Magic Quadrant Author Discusses Best Software and Buying Tips, *SearchDataManagement.com*, 13 June, available online at http://searchdatamanagement.techtarget.com/originalContent/0,289142,sid91_gci1260699,00.html.

SPSS (2007). SPSS Data Mining Tips, available online at http://www.spss.ch/upload/1124 797262_DMtips-Booklet.pdf.

Togaware (2007). Data Mining Catalogue, available online at http://www.togaware.com/ datamining/catalogue.html.

Worthen, M. (2005). Selecting the Ideal Business Intelligence/Data Analytics Solution, *DM Direct Special Report*, 12 April (Part 1) and 19 April (Part 2), available online at http://www.dmreview.com/article_sub.cfm?articleId=1025138 and http://www.dmrev iew.com/editorial/newsletter_article.cfm?nl=bireport&articleId=1025580&issue=20170).

KEY TERMS

Cross-Validation: Method of predictive model evaluation. It consists of splitting the available data into *N* subsets, or folds, of roughly equal size (and target feature distribution), successively building a predictive model from *N*-1 folds and testing on the remaining fold, making sure that each fold is used exactly once for testing. The model's overall performance is the average performance over all folds.

Data Mining: Application of visualization, statistics and machine learning to the discovery of knowledge in databases. There is general consensus that knowledge found by data mining should in some way be novel and actionable.

Dependency Model: Data Mining model that finds patterns of association among the components of aggregate data elements (e.g., market basket analysis). It is the result of unsupervised learning and its output generally takes the form of association rules.

Descriptive Model: Data Mining model that shows patterns of similarity among data elements (e.g., customer segmentation). It is the result of unsupervised learning and its output generally takes the form of clusters.

Predictive Model: Data Mining model that captures patterns of regularity between the characteristics of data elements and some specified target feature (e.g., credit risk assessment). It is the result of supervised learning and its output takes the form of a decision tree, decision list, probability model, neural network, etc.

ROC Analysis: Method of predictive model evaluation. It is based on the notion of Receiver Operating Characteristics (ROC) curve, where the evaluation consists not of a single accuracy value, but rather of a number of trade-off points (or curve) between false positive rate and true positive rate.

ENDNOTE

[1] The current database contains over 60 tools. It is updated on a regular basis so as to remain current, and is available freely from the author. Although it did not arise that way, this database can be viewed as a natural extension and update of the earlier list of Goebel & Gruenwald (1999).

Data Mining with Cubegrades

Amin A. Abdulghani
Data Mining Engineer, USA

INTRODUCTION

A lot of interest has been expressed in database mining using association rules (Agrawal, Imielinski, & Swami, 1993). In this chapter, we provide a different view of the association rules, referred to as *cubegrades* (Imielinski, Khachiyan, & Abdulghani, 2002) .

An example of a typical *association rule* states that, say, 23% of supermarket transactions (so called market basket data) which buy bread and butter buy also cereal (that percentage is called *confidence*) and that 10% of all transactions buy bread and butter (this is called *support*). Bread and butter represent the body of the rule and cereal constitutes the consequent of the rule. This statement is typically represented as a probabilistic rule. But *association rules* can also be viewed as statements about how the cell representing the body of the rule is affected by specializing it by adding an extra constraint expressed by the rule's consequent. Indeed, the confidence of an association rule can be viewed as the ratio of the support drop, when the cell corresponding to the body of a rule (in our case the cell of transactions buying bread and butter) is augmented with its consequent (in this case cereal). This interpretation gives association rules a "dynamic flavor" reflected in a hypothetical change of support affected by specializing the body cell to a cell whose description is a union of body and consequent descriptors. For example, our earlier association rule can be interpreted as saying that the count of transactions buying bread and butter drops to 23% of the original when restricted (rolled down) to the transactions buying bread, butter and cereal. In other words, this rule states how the count of transactions supporting buyers of bread and butter is affected by buying cereal as well.

With such interpretation in mind, a much more general view of association rules can be taken, when support (count) can be replaced by an arbitrary measure or aggregate and the specialization operation can be substituted with a different "delta" operation. Cubegrades capture this generalization. Conceptually, this is very similar to the notion of gradients used in calculus. By definition the gradient of a function between the domain points x_1 and x_2 measures the ratio of the *delta change* in the function value over the *delta change* between the points. For a given point x and function f(), it can be interpreted as a statement of how a change in the value of x (Δx), affects a change of value in the function ($\Delta f(x)$).

From another viewpoint, *cubegrades* can also be considered as defining a primitive for *data cubes*. Consider a 3-D cube model shown in Figure 1 representing sales data. It has three dimensions year, product and location. The measurement of interest is total sales. In olap terminology, since this cube models the base data, it forms a 3-D *base cuboid*. A *cuboid* in general is a group-by of a subset of dimensions of the base data, obtained by aggregating all tuples on these dimensions. So, for example for our sales data we have three 2-d cuboids namely (year, product), (product, location) and (year, location), three 1-d cuboids (year), (location) and (product) and one 0-d cuboid in which aggregation is performed on the whole data. For base data, with n diemensions, the union of all *k*-dimensional (k<=n) cuboids forms *an n-dimensional data cube*. A *cell* represents an association of a measure *m* (e.g., total sales) with a member of every dimension in a cuboid e.g. C1(product="toys", location="NJ", year="2004"). The dimensions not present in the cell are aggregated over all possible members. For example, you can have a two-dimensional (2-D) cell, C2(product="toys", year="2004"). Here, the implicit value for the dimension location is '*', and the measure *m* (e.g., total sales) is aggregated over all locations. Any of the standard aggregate functions such as count, total, average, minimum, or maximum can be used for aggregating. Suppose the sales for toys in 2004 for NJ, NY, PA were $2.5M, $3.5M, $1.5M respectively and that the aggregating function is total. Then, the measure value for cell C2 is $7.5M.

The scope of interest in OLAP is to evaluate one or more measure values of the cells in the cube. Cubegrades allow a broader, more dynamic view. In addition to evaluating the measure values in a cell,

Figure 1. A 3-D base cuboid with an example 3-D cell.

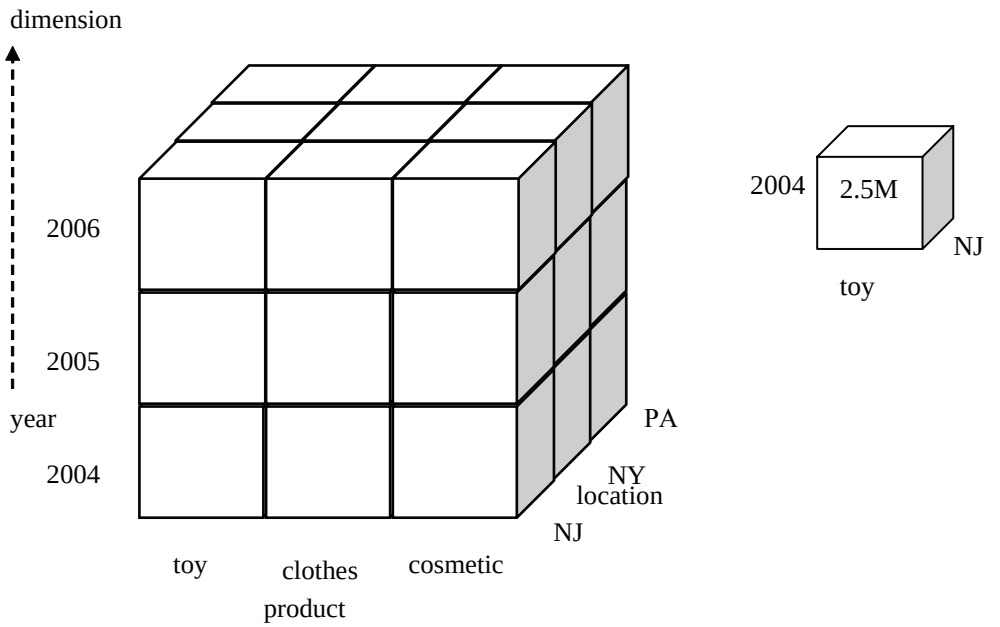
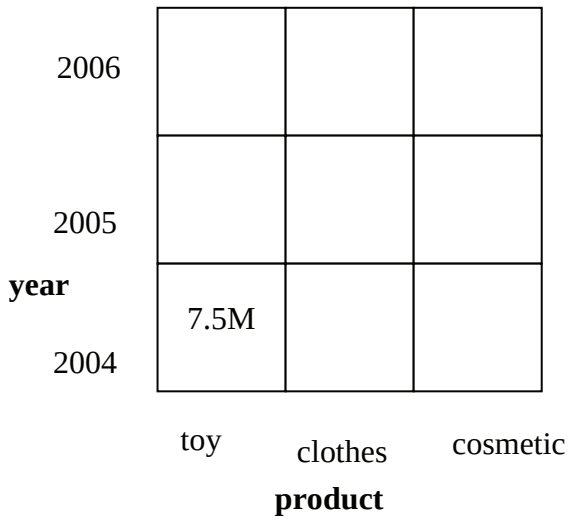


Figure 2. An example 2-D cuboid on (product, year) for the 3-D cube in Figure 1 (location=''); total sales needs to be aggregated (e.g., SUM)*



they evaluate how the measure values change or are affected in response to a change in the dimensions of a cell. Traditionally, OLAP have had operators such as drill downs, rollups defined, but the cubegrade operator differs from them as it returns a value measuring the effect of the operation. There have been additional operators proposed to evaluate/measure cell *interestingness* (Sarawagi, 2000; Sarawagi, Agrawal, & Megiddo, 1998). For example, Sarawagi et al., (1998) computes anticipated value for a cell using the neighborhood values, and a cell is considered an exception if its value is significantly different from its anticipated value. The difference is that cubegrades perform a direct cell to cell comparison.

BACKGROUND

An *association* or propositional rule can be defined in terms of cube cells. It can be defined as a quadruple *(body, consequent, support, confidence*) where *body* and *consequent* are cells over disjoint sets of attributes, *support* is the number of records satisfying the *body* and *confidence* is the ratio of the number of records which satisfy the *body* and the *consequent* to the number of records which satisfy just the body. We can also consider an association rule as a statement about a *"relative change"* of measure, COUNT, when "specializing" or "drilling down" the cell denoted by the *body* to the cell denoted by the *body* + *consequent*. The *confidence* of the rule measures how the *consequent* affects the support when drilling down the *body*. There are two ways these association rules can be generalized:

- By allowing relative changes in other measures, instead of just confidence, to be returned as part of the rule.
- By allowing cell modifications to be able to occur in different ``directions'' instead of just specializations (or drill-downs).

These generalized cell modifications are denoted as *cubegrades*. A cubegrade expresses how a change in the structure of a given cell affects a set of predefined measures. The original cell which is being modified is referred to as the *source* and the modified cell as *target*.

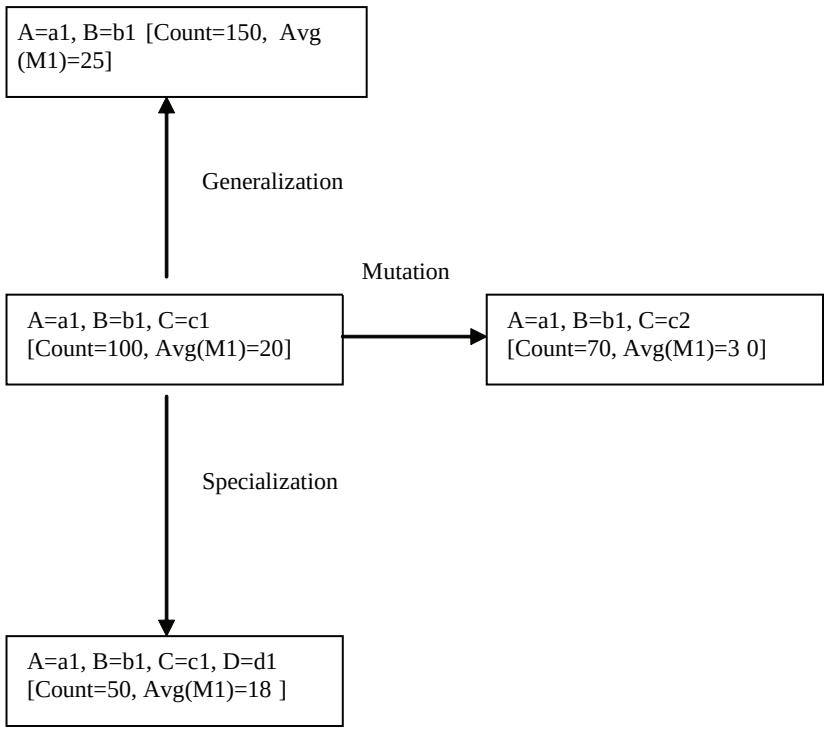
More formally, a cubegrade is a 5-tuple *(source, target, measures, value, delta-value)* where:.

- *source* and *target* are cube cells,
- *measures* is the set of measures which are evaluated both in the *source* as well as in the *target*,
- *value* is a function, value: *measures* → *R*, which evaluates measure *m* ∈ *measures* in the *source*,
- *delta-value* is also a function, *delta-value: measures* → *R,* which computes the ratio of the value of *m* ∈ *measures* in the *target* versus the *source.*

A cubegrade can visually be represented as a rule form:

Source → target, [measures, value, delta-value]

Figure 3. Cubegrade: Specialization, generalization and mutation



Define a *descriptor* to be an attribute value pair of the form *dimension=value* if the dimension is a discrete attribute *or dimension = [lo, hi]* if the attribute is a dimension attribute. We distinguish three types of the cubegrades:

- *Specializations:* A cubegrade is a *specialization* if the set of descriptors of the target are a superset of those in the source. Within the context of OLAP, the target cell is termed as a *drill-down* of *source.*
- *Generalizations:* A cubegrade is a *generalization* if the set of descriptors of the target cell are a subset of those in the source. Here, in OLAP, the target cell is termed as a *roll-up* of *source.*
- *Mutations:* A cubegrade is a *mutation* if the target and source cells have the same set of attributes but differ on the descriptor values (they are "union compatible" so to speak, as the term has been used in relational algebra).

Figure 2 illustrates the operations of these cubegrades. Following, we illustrate some specific examples to explain the use of these cubegrades

- (Specialization Cubegrade). The average sales of toys drops by 10% among buyers who live in PA.

  (product= "toy") → (product= "toy", location= "NJ")
  [AVG(sales),AVG(sales) = $85, DeltaAVG(sales) = 90%]

- (Mutation Cubegrade). The average sales drops by 30% when moving from NY buyers to PA buyers.

  (location= "NY") → (location= "PA")
  [AVG(sales), AVG(sales) = $110, DeltaAVG(sales)= 70%]

MAIN THRUST

Similar to association rules(Agrawal, & Srikant, 1994), the generation of *cubegrades* can be divided into two phases: (i) generation of significant cells (rather than frequent sets) satisfying the source cell conditions (ii) computation of cubegrades from the source (rather than computing association rules from frequent sets) satisfying the join conditions between source and target and target conditions.

The first task is similar to the computation of iceberg cube queries (Beyer & Ramakrishnan, 1999; Han, Pei, Dong, & Wang, 2001; Xin, Han, Li, & Wah, 2003). The fundamental property which allows for pruning in these computations is called *monotonicity* of the query: Let D be a database and $X \subseteq D$ be a cell. A query $Q(\cdot)$ is *monotonic* at X, if the condition Q(X) is FALSE implies Q(X') is FALSE for any $X` \subseteq X$.

However, as described by Imielinski et al (2002), determining whether a query Q is monotonic in terms of this definition is an NP-hard problem for many simple class of queries. To workaround this problem, the authors introduced another notion of monotonicity referred to *as view monotonicity* of a query. An important property of view monotonicity is that the time and space required for checking it for a query depends on the number of terms in the query and not the size of the database or the number of its attributes. Since most of the queries typically have few terms, it would be useful in many practical situations. The method presented can be used for checking for view monotonicity for queries that include constraints of type (Agg {<, >, =, !=} c) where c is a constant, and Agg can be MIN, SUM, MAX, AVERAGE and COUNT or an aggregate that is an higher order moment about the origin or an aggregate that is an integral of a function on a single attribute.

Consider a hypothetical query, asking for cells with 1000 or more buyers and with total milk sales less than $50,000. In addition, the average milk sales per customers should be between 20 to 50 dollars and with maximum sales greater than $75. This query can be expressed as follows:

COUNT(*)>=1000 and AVG(salesMilk)>=20 and AVG(salesMilk)<50 and MAX(salesMilk)>=75 and SUM(saleMilk)<50K

Suppose, while performing bottom up cube computation we have a cell C with the following view V (Count =1200; AVG(salesMilk) = 50; MAX(salesMilk) = 80; MIN(salesMilk) = 30; SUM(salesMilk) =60000). Then using the method for checking view monotonicity, it can be shown that there can exist some cell C' of C (though it's not guaranteed that this sub cell exists

in this database) with 1000<= count <1075 for which this query can be satisfied. Thus, this query can not be pruned on the cell.

However, if the view for C was (Count = 1200; AVG(salesMilk) = 57; MAX(salesMilk) =80; MIN(salesMilk) = 30; SUM(salesMilk) = 68400), then it can be shown that there doesn't exist any sub cell C' of C in any database for which the original query can be satisfied. Thus, the query can be pruned on cell C.

Once, the *source* cells have been computed, the next task is to compute the set of *target* cells. This is done by performing a set of query conversions which will make it possible to reduce cubegrade query evaluation to iceberg queries. Given a specific candidate source cell C, define Q[C] as the query which results from Q by *source substitution*. Here, Q is transformed by substituting into its "where" clause all *the values* of the *measures*, and the descriptors of the source cell C. This also includes performing the *delta elimination* step which replaces all the relative *delta values* (expressed as fractions) by the regular less than, greater than conditions. This is possible since the values for the *measure*s of C are known. With this, the *delta-values* can now be expressed as conditions on *values* of the *target*. For example, if AVG(Salary)=40K in cell C and DeltaAVG(Salary) is of the form DeltaAVG(Salary) >1.10, the transformed query would have the condition AVG(Salary) >44K where AVG(Salary) references the target cell. The final step in *source substitution* is *join transformation* where the join conditions (specializations, generalizations, mutations) in Q are transformed into the target conditions since the source cell is known. Notice, thus, that Q[C] is the cube query specifying for the target cell.

Dong, Han, Lam, Pei and Wang (2001) presents an optimized version of target generation which is particularly useful for the case where the number of source cells are few in numbers. The ideas in the algorithm include:

- Perform for the set of identified source cells the *lowest common delta elimination* such that the resulting *target condition* do not exclude any possible target cells.
- Perform a bottom-up iceberg query for the target cells based on the *target condition*. Define *LiveSet(T)* of a target cell T as the candidate set of source cells that can possibly match or join with target. A target cell, T, may identify a source, S, in its *LiveSet* to be prunable based on its monotonicity and thus removable from its *LiveSet.* In such a case, all descendants of T would also not include S in their *LiveSets*.
- Perform a join of the target and each of its *LiveSet's* source cells and for the resulting cubegrade, check if it satisfies the join criteria and the *Delta-Value* condition for the query.

In a typical scenario, it is not expected that users would be asking for cubegrades per se. Rather, it may be more likely that they pose a query on how a given delta change affects a set of cells, which cells are affected by a given delta change, or what delta changes affect a set of cells in a prespecified manner.

Further, more complicated set of applications can be implemented using cubegrades. For example, one may be interested in finding cells which remain stable and are not significantly affected by generalization, specialization or mutation. An illustration for that would be to find cells that remain stable on the blood pressure measure and are not affected by different specialization on age or area demographics.

FUTURE TRENDS

A major challenge for cubegrade processing is its computational complexity. Potentially, an exponential number of source/target cells can be generated. A positive development in this direction is the work done on Quotient Cubes (Lakshmanan, Pei, & Han, 2002). This work provides a method for partitioning and compressing the cells of a data cube into equivalent classes such that the resulting classes have cells that cover the same set of tuples and preserve the cube's semantic rollup/drilldown. In this context, we can reduce the number of cells generated for the source and target. Further, the pairings for the cubegrade can be reduced by restricting the source and target to belong to different classes. Another approach, recently proposed, for reducing the set of cubegrades to evaluate is to mine for the top-k cubegrades based on delta-values (Alves, R., Belo, O. & Ribeiro, J., 2007). The key additional pruning utilized there is to mine for regions in the cube that have high variance based on a given measure and thus provide a good candidate set for the set of target cells to consider for computing the cubegrades.

Another related challenge for cubegrades is to identify the set of interesting cubegrades (cubegrades that are somewhat surprising). Insights to this problem can be obtained from similar work done in the context of association rules (Bayardo & Agrawal, 1999; Liu, Ma, &Yu, 2001). Some progress has been made in direction with the extensions of the Lift criterion and Loevinger criterion to sum-based aggregate measures (Messaoud, Rabaseda, Boussaid, & Missaou, 2006).

CONCLUSION

In this chapter, we looked at a generalization of association rules referred to as cubegrades .These generalizations include allowing to evaluate relative changes in other measures, instead of just confidence, to be returned as well as allowing cell modifications to be able to occur in different "directions". The additional directions, we consider here, include generalizations which modifies cells towards the more general cell with fewer descriptors, and mutations, which modifies the descriptors of a subset of the attributes in the original cell definition with the others remaining the same. The paradigm allows us to ask queries that were not possible through association rules. The downside is that it comes with the price of relatively increased computation/storage costs that need to be tackled with innovative methods.

REFERENCES

Abdulghani, A. (2001). Cubegrades-Generalization of association rules to mine large datasets, PhD Thesis, Rutgers University, 2001.

Agrawal, R., Imielinski, T,. & Swami, A.N. (1993). Mining Association Rules between Sets of Items in Large Databases. SIGMOD 1993, 207-216.

Agrawal, R., & Srikant, R. (1994). Fast Algorithms for Mining Association Rules in Large Databases. VLDB 1994, 487-499.

Alves, R., Belo, O. & Ribeiro, J. (2007). Mining Top-K Multidimensional Gradients. Data Warehousing and Knowledge Discovery, Volume 4654/2000, Lecture Notes in Computer Science, 375-384.

Bayardo, R. & Agrawal R (1999). Mining the most interesting rules. KDD-99, 145-154.

Beyer, K.S., & Ramakrishnan R. (1999). Bottom-Up Computation of Sparse and Iceberg CUBEs**.** SIGMOD 1999, 359-370.

Dong, G., Han, J. Lam, J. M., Pei, J. & Wang, K. (2001). Mining Multi-Dimensional Constrained Gradients in Data Cubes. VLDB 2001, 321-330.

Gray, J., Bosworth, A., Layman, A., & Pirahesh, H. (1996). Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Total. ICDE 1996, 152-159.

Han, J., Pei, J., Dong, G., & Wang, K. (2001). Efficient computation of iceberg cubes with complex measures. SIGMOD 2001, 1-12.

Imielinski T., Khachiyan, L., & Abdulghani, A. (2002). Cubegrades: Generalizing Association Rules. Data Mining Knowledge Discovery 6(3). 219-257.

Lakshmanan, L.V.S., Pei, J., & Han, J. (2002). Quotient Cube: How to Summarize the Semantics of a Data Cube. VLDB 2002, 778-789.

Liu, B., Ma, Y., & Yu, S.P. (2001). Discovering unexpected information from your competitors' websites. KDD 2001, 144-153.

Messaoud, R., Rabseda, S., Bousssaid, O & Missaoui, R. (2006). Enhanced Mining of Association Rules from Data Cubes. DOLAP 2006, 11-18.

Sarawagi, S. (2000). User-Adaptive Exploration of Multidimensional Data. VLDB 2000, 307-316.

Sarawagi, S., Agrawal, R., & Megiddo, N.(1998). Discovery driven exploration of OLAP data cubes. *EDBT 1998*., 168-182.

Xin, D., Han, J., Li, X., & Wah, B. W.(2003). Star-Cubing: Computing Iceberg Cubes by Top-Down and Bottom-Up Integration**.** VLDB 2003, 476-487.

KEYTERMS

Cubegrade: A *cubegrade* is a 5-tuple (source, target, measures, value, delta-value) where:.

- source and target are cells,
- measures is the set of measures which are evaluated both in the source as well as in the target,
- value is a function, value: measures → R, which evaluates measure m ∈ measures in the source,
- delta-value is also a function, delta-value: measures → R, which computes the ratio of the value of m ∈ measures in the target versus the source

Drill-Down: A cube operation that allows users to navigate from summarized cells to more detailed cells.

Generalizations: A cubegrade is a *generalization* if the set of descriptors of the target cell are a subset of the set of attribute-value pairs of the source cell.

Gradient Cell: Synonym for a target cell of a cubegrade

Iceberg Cubes: Set of cells in a cube that satisfies an *iceberg query*.

Iceberg Query: A query on top of a cube that is asks for aggregates above a certain threshold.

Mutations: A cubegrade is a *mutation* if the target and source cells have the same set of attributes but differ on the values.

Query Monotonicity: A query $Q(\cdot)$ is *monotonic* at a cell X if the condition Q(X) is FALSE implies Q(X') is FALSE for any cell $X` \subseteq X$.

Roll-Up: A cube operation that allows users to aggregate from detailed cells to summarized cells.

Specialization: A cubegrade is a *specialization* if the set of attribute-value pairs of the target cell are a superset of the set of attribute-value pairs of the source cell.

View: A *view* V on S is an assignment of values to the elements of the set. If the assignment holds for the dimensions and measures in a given cell X, then V is a view for X on the set S.

View Monotonicity: A query Q is *view monotonic* on view V if for any cell X in any database D such that V is the view for X for query Q , the condition Q is FALSE for X implies Q is FALSE for all $X' \subseteq X$.

Data Mining with Incomplete Data

Hai Wang
Saint Mary's University, Canada

Shouhong Wang
University of Massachusetts Dartmouth, USA

INTRODUCTION

Survey is one of the common data acquisition methods for data mining (Brin, Rastogi & Shim, 2003). In data mining one can rarely find a survey data set that contains complete entries of each observation for all of the variables. Commonly, surveys and questionnaires are often only partially completed by respondents. The possible reasons for incomplete data could be numerous, including negligence, deliberate avoidance for privacy, ambiguity of the survey question, and aversion. The extent of damage of missing data is unknown when it is virtually impossible to return the survey or questionnaires to the data source for completion, but is one of the most important parts of knowledge for data mining to discover. In fact, missing data is an important debatable issue in the knowledge engineering field (Tseng, Wang, & Lee, 2003).

In mining a survey database with incomplete data, patterns of the missing data as well as the potential impacts of these missing data on the mining results constitute valuable knowledge. For instance, a data miner often wishes to know how reliable a data mining result is, if only the complete data entries are used; when and why certain types of values are often missing; what variables are correlated in terms of having missing values at the same time; what reason for incomplete data is likely, etc. These valuable pieces of knowledge can be discovered only after the missing part of the data set is fully explored.

BACKGROUND

There have been three traditional approaches to handling missing data in statistical analysis and data mining. One of the convenient solutions to incomplete data is to eliminate from the data set those records that have missing values (Little & Rubin, 2002). This, however, ignores potentially useful information in those records. In cases where the proportion of missing data is large, the data mining conclusions drawn from the screened data set are more likely misleading.

Another simple approach of dealing with missing data is to use generic "unknown" for all missing data items. However, this approach does not provide much information that might be useful for interpretation of missing data.

The third solution to dealing with missing data is to estimate the missing value in the data item. In the case of time series data, interpolation based on two adjacent data points that are observed is possible. In general cases, one may use some expected value in the data item based on statistical measures (Dempster, Laird, & Rubin, 1997). However, data in data mining are commonly of the types of ranking, category, multiple choices, and binary. Interpolation and use of an expected value for a particular missing data variable in these cases are generally inadequate. More importantly, a meaningful treatment of missing data shall always be independent of the problem being investigated (Batista & Monard, 2003).

More recently, there have been mathematical methods for finding the salient correlation structure, or aggregate conceptual directions, of a data set with missing data (Aggarwal & Parthasarathy, 2001; Parthasarathy & Aggarwal, 2003). These methods make themselves distinct from the traditional approaches of treating missing data by focusing on the collective effects of the missing data instead of individual missing values. However, these statistical models are data-driven, instead of problem-domain-driven. In fact, a particular data mining task is often related to its specific problem domain, and a single generic conceptual construction algorithm is insufficient to handle a variety of data mining tasks.

MAIN THRUST

There have been two primary approaches of data mining with incomplete data: conceptual construction and enhanced data mining.

Conceptual Construction with Incomplete Data

Conceptual construction with incomplete data reveals the patterns of the missing data as well as the potential impacts of these missing data on the mining results based only on the complete data. Conceptual construction on incomplete data is a knowledge development process. To construct new concepts on incomplete data, the data miner needs to identify a particular problem as a base for the construction. According to (Wang, S. & Wang, H., 2004), conceptual construction is carried out through two phases. First, data mining techniques (e.g., cluster analysis) are applied to the data set with complete data to reveal the unsuspected patterns of the data, and the problem is then articulated by the data miner. Second, the incomplete data with missing values related to the problem are used to construct new concepts. In this phase, the data miner evaluates the impacts of missing data on the identification of the problem and develops knowledge related to the problem. For example, suppose a data miner is investigating the profile of the consumers who are interested in a particular product. Using the complete data, the data miner has found that variable *i* (e.g., income) is an important factor of the consumers' purchasing behavior. To further verify and improve the data mining result, the data miner must develop new knowledge through mining the incomplete data. Four typical concepts as results of knowledge discovery in data mining with incomplete data are described as follows:

(1) **Reliability:** The reliability concept reveals the scope of the missing data in terms of the problem identified based only on complete data. For instance, in the above example, to develop the reliability concept, the data miner can define index $V_M(i)/V_C(i)$ where $V_M(i)$ is the number of missing values in variable *i*, and V_C(i) is the number of samples used for the problem identification in variable *i*. Accordingly, the higher $V_M(i)/V_C(i)$ is, the lower the reliability of the factor would be.

(2) **Hiding:** The concept of hiding reveals how likely an observation with a certain range of values in one variable is to have a missing value in another variable. For instance, in the above example, the data miner can define index $V_M(i)|x(j) \in (a,b)$ where $V_M(i)$ is the number of missing values in variable *i*, x(*j*) is the occurrence of variable *j* (e.g., education years), and (a,b) is the range of x(*j*); and use this index to disclose the hiding relationships between variables *i* and *j*, say, more than two thousand records have missing values in variable income given the value of education years ranging from 13 to 19.

(3) **Complementing:** The concept of complementing reveals what variables are more likely to have missing values at the same time; that is, the correlation of missing values related to the problem being investigated. For instance, in the above example, the data miner can define index $V_M(i,j)/V_M(i)$ where $V_M(i,j)$ is the number of missing values in both variables *i* and *j*, and $V_M(i)$ is the number of missing values in variable *i*. This concept discloses the correlation of two variables in terms of missing values. The higher the value $V_M(i,j)/V_M(i)$ is, the stronger the correlation of missing values would be.

(4) **Conditional Effects:** The concept of conditional effects reveals the potential changes to the understanding of the problem caused by the missing values. To develop the concept of conditional effects, the data miner assumes different possible values for the missing values, and then observe the possible changes of the nature of the problem. For instance, in the above example, the data miner can define index $\Delta P|\forall z(i)=k$ where ΔP is the change of the size of the target consumer group perceived by the data miner, $\forall z(i)$ represents all missing values of variable *i*, and k is the possible value variable *i* might have for the survey. Typically, k={*max*, *min*, *p*} where *max* is the maximal value of the scale, *min* is the minimal value of the scale, and *p* is the random variable with the same distribution function of the values in the complete data. By setting different possible values of k for the missing values, the data miner is able to observe the change of the size of the consumer group and redefine the problem.

Enhanced Data Mining with Incomplete Data

The second primary approach to data mining with incomplete data is enhanced data mining, in which incomplete data are fully utilized. Enhanced data mining is carried out through two phases. In the first phase, observations with missing data are transformed into fuzzy observations. Since missing values make the observation fuzzy, according to fuzzy set theory (Zadeh, 1978), an observation with missing values can be transformed into fuzzy patterns that are equivalent to the observation. For instance, suppose there is an observation $\boldsymbol{A}=\mathbf{X}(x_1, x_2, \ldots x_c \ldots x_m)$ where x_c is the variable with missing value, and $x_c \in \{ r_1, r_2 \ldots r_p \}$ where r_j (j=1, 2, ... p) is the possible occurrence of x_c. Let $\mu_j = P_j(x_c = r_j)$, the fuzzy membership (or possibility) that x_c belongs to r_j (j=1, 2, ...p), and $\sum_j \mu_j = 1$. Then, $\mu_j [\mathbf{X} | (x_c = r_j)]$ (j=1, 2, ... p) are fuzzy patterns that are the equivalence to the observation $\boldsymbol{A}$.

In the second phase of enhanced data mining, all fuzzy patterns, along with the complete data, are used for data mining using tools such as self-organizing maps (SOM) (Deboeck & Kohonen, 1998; Kohonen, 1989; Vesanto & Alhoniemi, 2000) and other types of neural networks (Wang, 2000, 2002). These tools used for enhanced data mining are different from the original ones in that they are capable of retaining information of fuzzy membership for each fuzzy pattern. Wang (2003) has developed a SOM-based enhanced data mining model to utilize all fuzzy patterns and the complete data for knowledge discovery. Using this model, the data miner is allowed to compare SOM based on complete data and fuzzy SOM based on all incomplete data to perceive covert patterns of the data set. It also allows the data miner to conduct what-if trials by including different portions of the incomplete data to disclose more accurate facts. Wang (2005) has developed a Hopfield neural network based model (Hopfield & Tank, 1986) for data mining with incomplete survey data. The enhanced data mining method utilizes more information provided by fuzzy patterns, and thus makes the data mining results more accurate. More importantly, it produces rich information about the uncertainty (or risk) of the data mining results.

FUTURE TRENDS

Research into data mining with incomplete data is still in its infancy. The literature on this issue is still scarce. Nevertheless, the data miners' endeavor for "knowing what we do not know" will accelerate research in this area. More theories and techniques of data mining with incomplete data will be developed in the near future, followed by comprehensive comparisons of these theories and techniques. These theories and techniques will be built on the combination of statistical models, neural networks, and computational algorithms. Data management systems on large-scale database systems for data mining with incomplete data will be available for data mining practitioners. In the long term, techniques of dealing with missing data will become a prerequisite of any data mining instrument.

CONCLUSION

Generally, knowledge discovery starts with the original problem identification. Yet the validation of the problem identified is typically beyond the database and generic algorithms themselves. During the knowledge discovery process, new concepts must be constructed through demystifying the data. Traditionally, incomplete data in data mining are often mistreated. As explained in this chapter, data with missing values must be taken into account in the knowledge discovery process.

There have been two major non-traditional approaches that can be effectively used for data mining with incomplete data. One approach is conceptual construction on incomplete data. It provides effective techniques for knowledge development so that the data miner is allowed to interpret the data mining results based on the particular problem domain and his/her perception of the missing data. The other approach is fuzzy transformation. According to this approach, observations with missing values are transformed into fuzzy patterns based on fuzzy set theory. These fuzzy patterns along with observations with complete data are then used for data mining through, for examples, data visualization and classification.

The inclusion of incomplete data for data mining would provide more information for the decision maker in identifying problems, verifying and improving the

data mining results derived from observations with complete data only.

REFERENCES

Aggarwal, C.C., & Parthasarathy, S. (2001). Mining massively incomplete data sets by conceptual reconstruction. *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 227-232). New York: ACM Press.

Batista, G., & Monard, M. (2003). An analysis of four missing data treatment methods for supervised learning. *Applied Artificial Intelligence*, *17*(5/6), 519-533.

Brin, S., Rastogi, R., & Shim, K. (2003). Mining optimized gain rules for numeric attributes. *IEEE Transactions on Knowledge & Data Engineering*, *15*(2), 324-338.

Deboeck, G., & Kohonen, T. (1998). *Visual explorations in finance with self-organizing maps*. London, UK: Springer-Verlag.

Dempster, A.P., Laird, N.M., & Rubin, D.B. (1997). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, B39*(1), 1-38.

Hopfield, J.J., & Tank, D. W. (1986). Computing with neural circuits. *Sciences, 233*, 625-633.

Kohonen, T. (1989). *Self-organization and associative memory* (3rd ed.). Berlin: Springer-Verlag.

Little, R.J.A., & Rubin, D.B. (2002). *Statistical analysis with missing data* (2nd ed.). New York: John Wiley and Sons.

Parthasarathy, S., & Aggarwal, C.C. (2003). On the use of conceptual reconstruction for mining massively incomplete data sets. *IEEE Computer Society*, *15*(6), 1512-1521.

Tseng, S., Wang, K., & Lee, C. (2003). A pre-processing method to deal with missing values by integrating clustering and regression techniques. *Applied Artificial Intelligence*, *17*(5/6), 535-544.

Vesanto, J., & Alhoniemi, E. (2000). Clustering of the self-organizing map. *IEEE Transactions on Neural Networks*, *11*(3), 586-600.

Wang, S. (2000). Neural networks. In M. Zeleny (Ed.), *IEBM handbook of IT in business* (pp. 382-391). London, UK: International Thomson Business Press.

Wang, S. (2002). Nonlinear pattern hypothesis generation for data mining. *Data & Knowledge Engineering*, *40*(3), 273-283.

Wang, S. (2003). Application of self-organizing maps for data mining with incomplete data Sets. *Neural Computing & Application*, *12*(1), 42-48.

Wang, S. (2005). Classification with incomplete survey data: A Hopfield neural network approach, *Computers & Operational Research, 32*(10), 2583-2594.

Wang, S., & Wang, H. (2004). Conceptual construction on incomplete survey data. *Data and Knowledge Engineering*, *49*(3), 311-323.

Zadeh, L.A. (1978). Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, *1*, 3-28.

KEY TERMS

Aggregate Conceptual Direction: Aggregate conceptual direction describes the trend in the data along which most of the variance occurs, taking the missing data into account.

Conceptual Construction with Incomplete Data: Conceptual construction with incomplete data is a knowledge development process that reveals the patterns of the missing data as well as the potential impacts of these missing data on the mining results based only on the complete data.

Enhanced Data Mining with Incomplete Data: Data mining that utilizes incomplete data through fuzzy transformation.

Fuzzy Transformation: The process of transforming an observation with missing values into fuzzy patterns that are equivalent to the observation based on fuzzy set theory.

Hopfield Neural Network: A neural network with a single layer of nodes that have binary inputs and outputs. The output of each node is fed back to all other nodes simultaneously, and each of the node forms a weighted sum of inputs and passes the output result through a

nonlinearity function. It applies a supervised learning algorithm, and the learning process continues until a stable state is reached.

Incomplete Data: The data set for data mining contains some data entries with missing values. For instance, when surveys and questionnaires are partially completed by respondents, the entire response data becomes incomplete data.

Neural Network: A set of computer hardware and/or software that attempt to emulate the information processing patterns of the biological brain. A neural network consists of four main components:

(1) Processing units (or neurons); and each of them has a certain output level at any point in time.
(2) Weighted interconnections between the various processing units which determine how the output of one unit leads to input for another unit.
(3) An activation rule which acts on the set of input at a processing unit to produce a new output.
(4) A learning rule that specifies how to adjust the weights for a given input/output pair.

Self-Organizing Map (SOM): Two layer neural network that maps the high-dimensional data onto low-dimensional grid of neurons through unsupervised learning or competitive learning process. It allows the data miner to view the clusters on the output maps.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 293-296, copyright 2005 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

Data Pattern Tutor for AprioriAll and PrefixSpan

Mohammed Alshalalfa
University of Calgary, Canada

Ryan Harrison
University of Calgary, Canada

Jeremy Luterbach
University of Calgary, Canada

Keivan Kianmehr
University of Calgary, Canada

Reda Alhajj
University of Calgary, Canada

INTRODUCTION

Data mining can be described as data processing using sophisticated data search capabilities and statistical algorithms to discover patterns and correlations in large pre-existing databases (Agrawal & Srikant 1995; Zhao & Sourav 2003). From these patterns, new and important information can be obtained that will lead to the discovery of new meanings which can then be translated into enhancements in many current fields.

In this paper, we focus on the usability of sequential data mining algorithms. Based on a conducted user study, many of these algorithms are difficult to comprehend. Our goal is to make an interface that acts as a "tutor" to help the users understand better how data mining works. We consider two of the algorithms more commonly used by our students for discovering sequential patterns, namely the AprioriAll and the PrefixSpan algorithms. We hope to generate some educational value, such that the tool could be used as a teaching aid for comprehending data mining algorithms. We concentrated our effort to develop the user interface to be easy to use by naïve end users with minimum computer literacy; the interface is intended to be used by beginners. This will help in having a wider audience and users for the developed tool.

BACKGROUND

Kopanakis and Theodoulidis (2003) highlight the importance of visual data mining and how pictorial representation of data mining outcomes are more meaningful than plain statistics, especially for non-technical users. They suggest many modeling techniques pertaining to association rules, relevance analysis, and classification. With regards to association rules they suggest using grid and bar representations for visualizing not only the raw data but also support, confidence, association rules, and evolution of time.

Eureka! is a visual knowledge discovery tool that specializes in two dimensional (2D) modeling of clustered data for extracting interesting patterns from them (Manco, Pizzuti & Talia 2004). VidaMine is a general purpose tool that provides three visual data mining modeling environments to its user: (a) the meta-query environment allows users through the use of "hooks" and "chains" to specify relationships between the datasets provided as input; (b) the association rule environment allows users to create association rules by dragging and dropping items into both the IF and THEN baskets; and (c) the clustering environment for selecting data clusters and their attributes (Kimani, *et al.*, 2004). After the model derivation phase, the user can perform analysis and visualize the results.

MAIN THRUST

AprioriAll is a equential data pattern discovery algorithm. It involves a sequence of five phases that work together to uncover sequential data patterns in large datasets. The first three phases, Sorting, L-itemset, and Transformation, take the original database and prepare the information for AprioriAll. The Sorting phase begins by grouping the information, for example a list of customer transactions, into groups of sequences with customer ID as a primary key. The L-itemset phase then scans the sorted database to obtain length one itemsets according to a predetermined minimum support value. These length one itemsets are then mapped to integer value, which will make generating larger candidate patterns much easier. In the Transformation phase, the sorted database is then updated to use the mapped values from the previous phase. If an item in the original sequence does not meet minimum support, it is removed in this phase, as only the parts of the customer sequences that include items found in the length one itemsets can be represented.

After preprocessing the data, AprioriAll efficiently determines sequential patterns in the Sequence phase. Length K sequences are used to generate length K+1 candidate sequences until K+1 sequences can no longer be generated (i.e., K+1, is greater than the largest sequence in the transformed database. Finally, the Maximal Phase prunes down this list of candidates by removing any sequential patterns that are contained within a larger sequential pattern.

Although this algorithm produces the desired results, it has several disadvantages. The potential for huge sets of candidate sequences is a major concern (Pei, *et al*., 2001). This results in an immensely large amount of memory space being used, especially when databases contain several large sequences. Another disadvantage is the time required to process large datasets since the algorithm requires multiple passes over the database. Additionally, AprioriAll has some difficulty mining long sequential patterns.

PrefixSpan requires preprocessing the database into a database consisting of sequences. The initial step scans the database and finds all length-1 prefixes that meet the minimum support. Next, the search space is partitioned into chunks corresponding to each length-1 prefix found in the previous step. Finally, the subsets of each of the prefixes can be mined by constructing corresponding projected databases; then each will be mined recursively. The final result is a compiled table consisting of the prefixes, postfixes, and all sequential patterns generated by the algorithm.

The major cost of PrefixSpan is the cost of construction of all the projected databases. In its worst case, it will construct a projected database for every sequential pattern (Pei, *et al*., 2001). "If the number and/or the size of projected databases can be reduced, the performance of sequential data mining can be improved substantially." (Pei, *et al*., 2001). One solution is the Bi-Level Projection. The Bi-Level Projection is represented in a lower triangular matrix, which can be used to generate the candidate sequential patterns.

The data mining technology is becoming increasingly popular and attractive as prediction technique, especially for advanced and emerging challenging applications. To compensate for this trend, it is crucial that more time be spent in the teaching process of many data mining algorithms. However, current resources that could serve as a learning aid lack the necessary tools to successfully present the material in an easy to understand manner. Taking a closer look at both AprioriAll and PrefixSpan, we found that they are not as straight forward as one might have liked. Needless to say, our experience with beginners tells that someone who decides to learn these algorithms may be left frustrated and confused. On a positive note, a visual interface which could be used as a tutor is an ideal solution to overcome the frustration and confusion. Compiling as many of the current resources and ideas regarding these algorithms into one user friendly visual interface seems like a good start. We decided to start from the bottom up. This means a systematical completion of the entire design process. With some basic background in Human Computer Interaction principles, we implemented a design that is best suited to be a potential ideal learning tool for both algorithms.

While designing our interface we implemented several features that allow for easier learning. The first implemented feature is to provide to the user all the steps of the algorithm while giving the user the option to selectively choose what step they want to display. Another feature of our interface is the clear textural description that couples with the selected step. Our system allows for some aspects of the interface to be interactive and better demonstrate how an algorithm works. Our interface also allows for a slideshow type feature to consecutively traverse all steps in an algorithm. A feature that we felt quite useful is to include

help features to aid a user who needs more than a general description of each algorithm. The introduction of color in the tables gives a better understanding of what is happening in each step. Allowing the user to output results based on the run-though of our interface is a necessary feature to allow the user to review at their own leisure. All of these features add to the ease of learning of the complex data mining algorithms.

Introducing the algorithms in an easy to understand manner is a vital aspect of our design. To accomplish this goal, we want to allow each part of the algorithm to be displayed when the interface is initially loaded. Since each algorithm has several steps or phases, an efficient way to present the numerous steps is very important. We decided to implement an "Aqua-Doc" style tool bar; this concept is borrowed from the Mac OS (see Figure 1). Each step of the algorithm is represented in small icon image and when the cursor passes over one of these icons, it magnifies that icon and half magnifies the icons direct neighbors. This allows a user to move the cursor slowly over each icon to navigate through the steps or phases. The interface initially boots up into a starting state. This starting state begins by putting all of the icons in a neutral state, with the main view of the current step and description set to the first step. By doing this, we are giving the user a place to begin, and giving them the overall feel of a tutorial beginning. When a user clicks or selects a certain step, that step is then projected into the main focus area with a clear textual description of the step directly beside it (see Figure 2). In introducing the steps in this fashion, a user will have a greater understanding of the complex algorithm, and can easily navigate through the said algorithm easily and efficiently.

When it comes to writing the textual descriptions that are used in correlation with each step, we tried to come up with a simple explanation of what is occurring in the step. This description is a key in the understanding of an algorithm. So by looking at a collection of previously written descriptions of a step and either

Figure 1. A view of the interactive tool bar that uses the Mac OS inspired "Aqua-Doc"

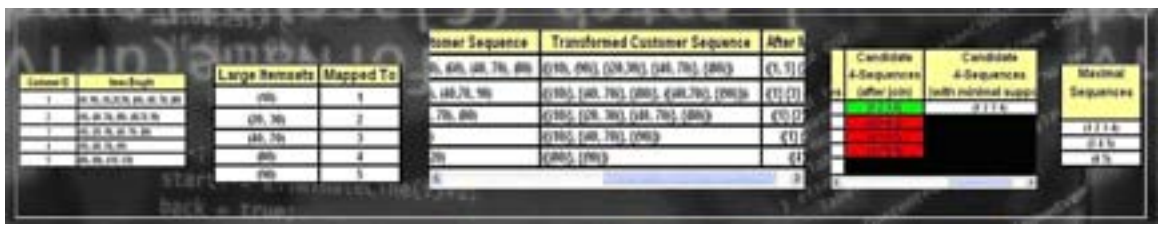
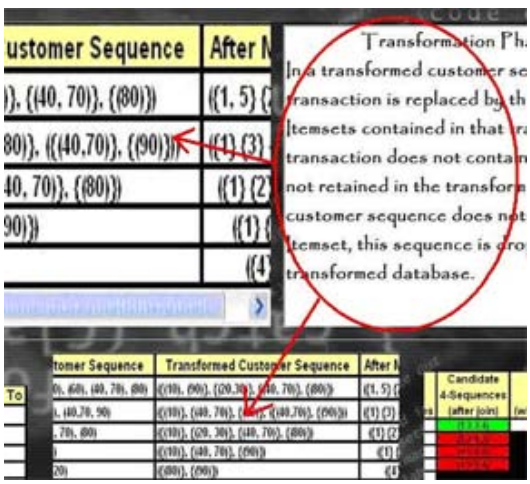


Figure 2. Textual description is generated by clicking on the interactive tool bar, and corresponds to the current step in the main focus area



taking the most understandable description or rewriting the description, we are successfully able to present the description efficiently.

With a learning tool, it is essential that you allow the user to interact with the data in a way that enhances their understanding of the material. We had two main areas within our interface that would allow the performance of the algorithms to be manipulated to the benefit of the user (see Figure 3). The first is the ability to change the minimum support. This can prove to be very advantageous for adjusting the amount and the overall size of sequential patterns used. Typically it starts as a small value, which will produce smaller and less complex sequential patterns. The second area that can be modified is the percent rules to show. When dealing with large datasets, the final resulting number of rules can be increasingly large. With so many rules visible to the user, the final result could prove to be somewhat overwhelming. If a user is just starting to understand an algorithm, just a few simple results are more than enough to develop a basic understanding.

Another helpful feature of our interface is the use of a slideshow feature. The actual toolbar has been developed using the Windows Media Player toolbar as a foundation, and has been altered to meet the needs of our slideshow toolbar. If the user selects play, the steps of the algorithm will be traversed using a predetermined amount of time per step. Users still have the opportunity to use the interactive toolbar on the bottom of the interface to override the slideshow. As the slideshow moves between each step, the interactive tool bar will also procedurally traverse the steps in direct correlation to the current step displayed by the slideshow. This is demonstrated to the users, where they are in the overall algorithm. The ability to move back and forth between algorithms using the tab structure can also pause the current slideshow, allowing for a "current state" of the interface to be temporarily saved (see Figure 4).

By showing exactly how the results for L-itemset phase in AprioriAll are determined, a user would have a much better understanding of the underlying processes. We display all potential length one itemsets in a color coordinated table (see Figure 5). All itemsets that meet the minimum support will be colored green, and the other rejected itemsets are colored red. The color scheme mentioned above has been also implemented in other areas of the interface corresponding to steps that preform any kind of removal from the final result. In the PrefixSpan algorithm, the final result is displayed

Figure 3. Drop down boxes for changing minimum support and percent of rules to show

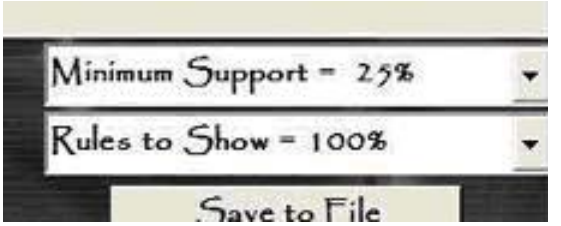
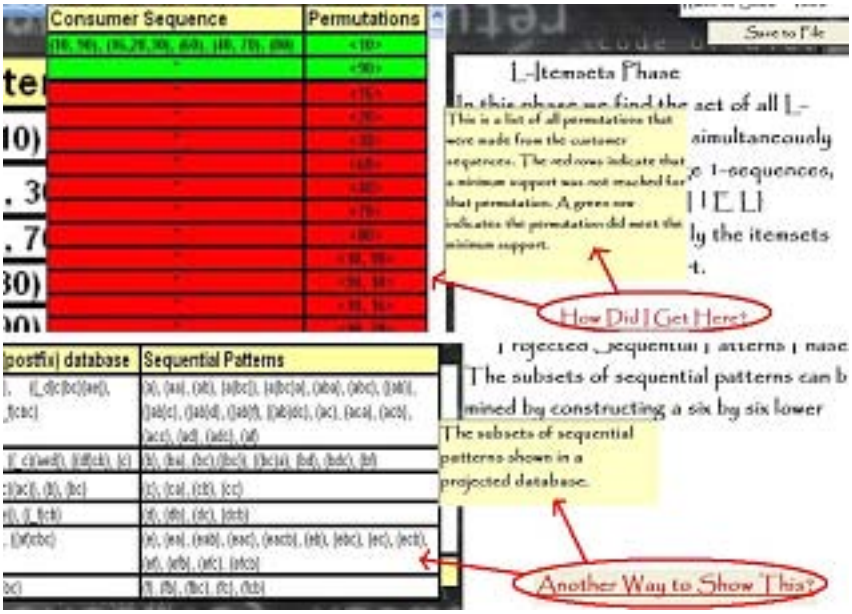


Figure 4. The tab for users to Pause, Play, Stop, and skip ahead or back a step through the slide show

Figure 5. Two instances of the help step being used. The top implements the "How Did I Get Here" feature while the bottom uses the "Another Way to Show This?" feature



in the matrix view. Just looking at this view initially can be slightly confusing. Here the option to show the results in a different format adds to the overall appeal of our design (see Figure 5).

Finally, we have added the functionality to be able to output the resulting sequential patterns to a file. This gives an additional benefit to users because they are then able to go back and look at this file to review what they have learned. The minimum support and percentage of rules to show are linked directly to this as the output file will be a direct representation of what the interactive attributes in the interface have been set to. The output file is also formatted to easily get useful information from.

Some of the highlighting points of our system that are clear benefits to students are the advantage of using continuing examples and the ability to interact with examples to make them easier to follow. In many of the resources we investigated, it is very uncommon for an example to use the same data from start to end. Our interface makes better understandable how an algorithm works much more practical as the example the user started with is the one that he/she has results for throughout the process. As well, being able to manipulate the current example, allows users to begin with a small set of data and slowly work their way up to a complete example. When a user wants to go back and review the material, the option to have previously saved output file is a major advantage. This cuts down on the time it would take to redo the algorithm example, especially because the output file that was previously saved based on the customized dataset (using results to show and minimum support).

We have already tested the usage and friendliness of the tool described in this paper on two different groups of students. The two groups were first given a tutorial about the general usage of the tool presented in this paper without any description of the algorithms; only they were introduced to the general usage of the interface. Then, one of the groups was first asked to learn the two algorithms using the visual tool presented in this paper and then the students were introduced to the algorithms in a classroom setting. On the other hand, the opposite was done with the other group, i.e., the students were first introduced to the algorithms in the classroom and then the students were asked to practice using the visual tool presented in this paper. It was surprisingly found out that most of the students (close to 95%) who were asked to use the tool first learned the algorithms faster and found it easier to follow the

material in the classroom. From the other group, only few students (close to 15%) enjoyed the way they followed in learning the algorithms first in class then practicing using the described tool.

FUTURE TRENDS

As discussed above, our interface only encompasses two of the many sequential data pattern algorithms. While this is an excellent start, it is a small portion of all the data mining algorithms out there. If we could expand our system to handle more of the algorithms, this would only increase the educational value of our tool and would make it a viable asset to any data mining classroom environment.

We would also like to introduce the capability of using datasets of arbitrary size for the examples, i.e., increase the level of difficulty where the user is expected to follow the mining process. Currently, small datasets are expected to be used to run though each of the algorithms. While this is presently a necessity to demonstrate each of the algorithms in our interface, the ability to explore any dataset would be much more beneficial to a user after building some background with the two algorithms. Users would then be able to create and run through any dataset that they would find useful to them. This would add immensely to the learning process as a user would not be restricted to use small datasets. However, expanding to larger datasets requires using a sampling technique to decide on parts of the output of each step to be displayed on the screen, and also requires the capability to save the whole intermediate results for later manual check of them. On the other hand, by creating a unique dataset, the perception of how datasets are generated is also added to the understanding of the data mining process.

As an added feature, the ability to have online support would be helpful to students who still have some hard questions that our interface does not answer. Using either an online tutor service via an email correspondence or an online discussion board, we feel that the needs of students using the system could definitely be satisfied.

CONCLUSION

In retrospect, the importance of data mining in the current industry is becoming more apparent. With this ever increasing demand for algorithms and systems to effectively deliver usable results, the fundamentals must first be reviewed. Learning how exactly data mining algorithms work will be crucial to understanding how they can be applied to the industry.

Given that AprioriAll and PrefixSpan are well known algorithms, they have been excellent methods for us to start with. Both demonstrate the basics of how data mining algorithms discover sequential data patterns, which make the overall appeal of our system a good one. However, with more time, our system can easily be expanded to handle even more algorithms. This in turn will add to the overall educational value that our system provides. The current system successfully meets the initial goals of our project. With a better understanding of how data mining works, the potential for young minds to create better more efficient algorithms and data mining tools is amplified to the level required by the current industry. As a result, the technological world as a whole will serve to benefit.

REFERENCES

Agrawal R., & Srikant R. (1995). Mining sequential patterns. *Proc. of IEEE ICDE*, pp.3-14.

Antunes C., & Oliveira, A. (2004). Generalization of pattern-growth methods for sequential pattern mining with gap constraints. *Proc. of Machine Learning and Data Mining in Pattern Recognition*, pp. 239-251.

Kuba P., & Popelínský, L. (2004). Mining frequent patterns in object-oriented data. *Proc. of the Workshop on Mining Graphs, Trees and Sequences,* pp. 15-25.

Pei J. et al., (2001). PrefixSpan: Mining sequential patterns efficiently by prefix projected pattern growth. *Proc. of IEEE ICDE*, pp. 215-224.

Mannila H., Toivonen H., & Verkamo, A. I. (1997). Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*, *1*(3), 259-289.

Zhao, Q., & Sourav, S.B. (2003). *Sequential pattern mining: A survey.* Technical Report. School of Computer Engineering, Nanyang Technological University, Singapore.

Kimani, S. et al., (2004). VidaMine: A visual data mining environment. *Visual Languages and Computing*, *15*(1), 37–67.

Kopanakis, I., & Theodoulidis, B. (2003). Visual data mining modeling techniques for the visualization of mining outcomes. *Visual Languages and Computing, 14*(6), 543–589.

Manco, G., Pizzuti, C., & Talia, D. (2004). Eureka!: An Interactive and Visual Knowledge Discovery Tool. *Visual Languages and Computing*, *15*(1), 1–35.

Windows Media Player 10, Microsoft Inc.

KEY TERMS

Association Rule: Generally represented as a correlation X→Y, where X and Y are two no-empty and disjoint sets of items; X is called antecedent, Y is called consequence.

Confidence: Measures the rule's strength; the rule X→Y has confidence *c* if *c*% of transactions in the database that contain X also contain Y.

Data Mining: Extraction of non-trivial, implicit, previously unknown and potentially useful information or patterns from data in large databases.

Frequent Itemset: An itemset is said to be frequent if its support is greater than a predefined threshold (minimum support value).

Itemset: A collection of one or more items; a subset of the items that appear in a given dataset.

Support: Indicates the frequencies of the occurring patterns; rule X→Y has support *s* if *s*% of transactions in the database contain X∪Y.

User Interface: A mode of interaction provided for the user to easily communicate with a given program or system.

Data Preparation for Data Mining

Magdi Kamel
Naval Postgraduate School, USA

INTRODUCTION

Practical experience of data mining has revealed that preparing data is the most time-consuming phase of any data mining project. Estimates of the amount of time and resources spent on data preparation vary from at least 60% to upward of 80% (SPSS, 2002a). In spite of this fact, not enough attention is given to this important task, thus perpetuating the idea that the core of the data mining effort is the modeling process rather than all phases of the data mining life cycle. This article presents an overview of the most important issues and considerations for preparing data for data mining.

BACKGROUND

The explosive growth of government, business, and scientific databases has overwhelmed the traditional, manual approaches to data analysis and created a need for a new generation of techniques and tools for intelligent and automated knowledge discovery in data. The field of knowledge discovery, better known as data mining, is an emerging and rapidly evolving field that draws from other established disciplines such as databases, applied statistics, visualization, artificial intelligence and pattern recognition that specifically focus on fulfilling this need. The goal of data mining is to develop techniques for identifying novel and potentially useful patterns in large data sets (Tan, Steinbach, & Kumar, 2006).

Rather than being a single activity, data mining is as an interactive and iterative process that consists of a number of activities for discovering useful knowledge (Han & Kamber, 2006). These include data selection, data reorganization, data exploration, data cleaning, and data transformation. Additional activities are required to ensure that useful knowledge is derived from the data after data-mining algorithms are applied such as the proper interpretation of the results of data mining.

MAIN FOCUS

In this article, we address the issue of data preparation—how to make the data more suitable for data mining. Data preparation is a broad area and consists of a number of different approaches and techniques that are interrelated in complex ways. For the purpose of this article we consider data preparation to include the tasks of data selection, data reorganization, data exploration, data cleaning, and data transformation. These tasks are discussed in detail in subsequent sections.

It is important to note that the existence of a well designed and constructed data warehouse, a special database that contains data from multiple sources that are cleaned, merged, and reorganized for reporting and data analysis, may make the step of data preparation faster and less problematic. However, the existence of a data warehouse is not necessary for successful data mining. If the data required for data mining already exist, or can be easily created, then the existence of a data warehouse is immaterial.

Data Selection

Data selection refers to the task of extracting smaller data sets from larger databases or data files through a process known as sampling. Sampling is mainly used to reduce overall file sizes for training and validating data mining models (Gaohua & Liu, 2000). Sampling should be done so that the resulting smaller dataset is representative of the original, large file. An exception of this rule is when modeling infrequent events such as fraud or rare medical conditions. In this case oversampling is used to boost the cases from the rare categories for the training set (Weiss, 2004). However, the validation set must approximate the population distribution of the target variable.

There are many methods for sampling data. Regardless to the type of sampling it is important to construct

a *probability sample*, one in which each record of the data has a known probability of being included in the sample. The importance of using probability samples is that it allows us to calculate a level of error on the statistics calculated from the sample data when using statistical inferential techniques.

The two main types of sampling include simple random sampling and stratified random sampling. In *simple random sampling*, each record in the original data file has an equal probability of being selected. In *stratified random sampling*, records are selected such that they are proportional to the segments of population they represent. A similar type of sampling is sampling over time. In this type, samples are selected so as to adequately represent all time periods of interest.

Another aspect of data selection is the selection of a subset of variables, an approach knows as dimensionality reduction (Liu & Motoda, 1998). With a smaller number of variables many data mining algorithms work better, and the resulting models are more understandable.

As data is being prepared, it is usually difficult to determine which variables are likely to be important for data mining. A good understanding of the business and the goals of data mining should provide at this stage a general idea on what variables might be important and therefore should be included in the analysis.

Data Reorganization

Data reorganization is used to change the case basis of a data set. Data can be reorganized through summarization or other complex file manipulation. Summarization replaces the current dataset with one containing summary information based on one or more subgroups. For example, a bank transactions dataset can be summarized by customer, account type, or branch.

Complex file manipulation tasks can include one or many types of operations, such as matching, appending, aggregating, and distributing data from one variable into several other new variables (Bock & Diday, 2000).

It is important to note that data needs to be explored, cleaned and checked before it is summarized. The reason for doing so is that failure to eliminate errors at a lower level of summarization will make it impossible to find at a higher level of summarization.

Data Exploration

Data exploration is the preliminary investigation of the data in order to better understand its characteristics. There is nothing particularly distinctive about data exploration for data mining than that used for Exploratory Data Analysis (EDA), which was created by John Tukey in the 1970's (1977). It usually involves the following tasks:

1. Examining the distribution of each variable, summary statistics, and the amount and type of missing data. This can be accomplished by creating frequencies table and/or appropriate graphs such as bar charts and histograms.
2. Studying the relationship between two variables of interest using techniques such as crosstabulations, correlations, and On-Line Analytical Processing (OLAP), as well as graphs such as clustered bar charts, scatterplots, and web graphs.
3. Determining the extent and patterns of missing data.

We briefly examine three major areas of data exploration: summary statistics, data visualization, and OLAP.

Summary Statistics

Summary statistics capture many characteristics of large set of values with a single number or a small set of numbers (Devore, 2003). For categorical data, summary statistics include the *frequency* which is the number of times each value occurs in a particular set of data, and the *mode* which is the value that has the highest frequency. For continuous data, the most widely used summary statistics are the mean and median, which are measures of central tendency. The *mean* is the average value of a set of values, while the *median* is the middle value if there are an odd number of values and the average of the two middle values if the number of values is even.

Another set of commonly used summary statistics for continuous data measures the variability or spread of a set of values. These measures include the range, the standard deviation, and the variance. The *range* is the difference between the highest and lowest values in a

set of numbers. The *standard deviation* is a measure of the average deviation from the mean, and the *variance* is the square of the standard deviation.

For multivariate data with continuous data, summary statistics include the covariance and correlation matrices. The *covariance* of two attributes is a measure of the degree to which two attributes vary together and depends on the magnitude of the variables. The *correlation* of two attributes is a measure of how strongly two attributes are linearly related. It is usually preferred to covariance for data exploration.

Other types of summary statistics include the *skewness* which measures the degree to which a set of values is symmetrically distributed around the mean, and the *modality* which indicates whether the data has multiple "bumps" where a majority of the data is concentrated.

Data Visualization

Data Visualization is the display of data in tabular or graphic format. A main motivation for using visualization is that data presented in visual form is the best way of finding patterns of interest (SPSS, 2002b). By using domain knowledge, a person can often quickly eliminate uninteresting patterns and focus on patterns that are important.

There are numerous ways to classify data visualization techniques (Fayyad, Grinstein, & Wierse, 2001). One such classification is based on the number of variables involved (one, two, or many). Another classification is based on the characteristics of the data, such as graphical or graph structure. Other classifications are based on the type of variables involved, and the type of application: scientific, business, statistical, etc. In the following section, we discuss the visualization of data with a single variable and with two variables in line with our approach described in at the beginning of the Data Exploration Section.

Visualization of Single Variables

There are numerous plots to visualize single variables. They include bar charts, histograms, boxplots, and pie charts. A *bar chart* is a graph that shows the occurrence of categorical values, while a *histogram* is a plot that shows the occurrence of numeric values by dividing the possible values into bins and showing the number of cases that fall into each bin. Bar charts and histograms are frequently used to reveal imbalances in the data.

Boxplots indicate the 10^{th}, 25^{th}, 50^{th}, 75^{th}, and 90^{th} percentiles of single numerical variables. They also show outliers which are shown by "+" marks. *Pie charts* are used with categorical variables that have a relatively small number of values. Unlike a histogram, a pie chart uses the relative area of a circle to indicate relative frequency.

Visualization of Two Variables

Graphs for visualizing the relationship between two variables include clustered bar charts and histograms, scatter plots, and Web charts. C*lustered bar charts* and *histograms* are used to compare the relative contribution of a second variable on the occurrence of a first variable.

A *scatter plot* shows the relationship between two numeric fields. Each case is plotted as a point in the plane using the values of the two variables as *x* and *y* coordinates. A *Web chart* illustrates the strength of the relationship between values of two (or more) categorical fields. The graph uses lines of various widths to indicate connection strength.

OLAP

On-Line Analytical Processing (OLAP) is an approach that represents data sets as multidimensional arrays (Ramakrishnan & Gehrke, 2002). Two main steps are necessary for this representation: 1) the identification of the dimensions of the array, and 2) the identification of a variable that is the focus of the analysis. The dimensions are categorical variables or continuous variables that have been converted to categorical variables. The values of the variable serve as indices into the array for the dimension corresponding to the variable. Each combination of variable values defines a cell of the multidimensional array. The content of each cell represents the value of a target variable that we are interested in analyzing. The target variable is a numerical field that can be aggregated and averaged.

Once a multidimensional array, or OLAP cube, is created, various operations could be applied to manipulate it. These operations include slicing, dicing, dimensionality reduction, roll up, and drill down. *Slicing* is selecting a group of cells from the entire multidimen-

sional array by specifying a specific value for one or more dimension variables. *Dicing* involves selecting a subset of cells by specifying a range of attribute values. *Dimensionality reduction* eliminates a dimension by summing over it, and is similar to the concept of aggregation discussed earlier. A *roll-up* operation performs aggregation on the target variable of the array either by navigating up the data hierarchy for a dimension or by dimensionality reduction. A *drill-down* operation is the reverse of roll-up. It navigates the data hierarchy from less detailed data to more detailed data.

Data Cleaning

Real-world data sources tend to suffer from inconsistency, noise, missing values, and a host of other problems (Berry & Linoff, 2004; Olson, 2003). Data cleaning (or data cleansing) techniques attempt to address and resolve these problems. Some common problems include:

a. *Inconsistent values of same variables in different datasets* - Marital status could be represented as "S", "M", and "W" in one data source and "1", "2", "3" in another. Misspelling of text data in some data sources is another example of this type of error
b. *Errors or noise in data* - This problem refers to fields that contain values that are incorrect. Some errors are obvious, such as a negative age value. Others are not as obvious, such as an incorrect date of birth.
c. *Outliers and skewed distributions* - Although not errors in the data, outliers are values of a variable that are unusual with respect to the typical values for that variable and can bias the analysis of data mining. Similarly skewed distributions may lead to lower model accuracy of the data mining algorithm.
d. *Missing values* - These refer to data that should be present but is not. In some cases, the data was not collected. In other cases, the data is not applicable to a specific case. Missing values are usually represented as NULL values in the data source. Data mining tools vary in their handling of missing value, but usually do not consider the records with missing data. NULL values may however represent acceptable values and therefore need to be included in the analysis.
e. *Missing cases* - This problem occurs when records of a segment of interest are completely missing from the data source. In this case the models developed will not apply to the missing segment.

In the next section we discuss ways of handling missing values, and in the following section, we discuss a general process for data cleaning.

Missing Values

Missing values are a most subtle type of data problem in data mining because the effect is not easily recognized. There are at least two potential problems associated with missing values. The first relates to sample size and statistical power, the second to bias.

If the amount of missing values is large, the number of valid records in the sample will be reduced significantly, which leads to loss of statistical power. On the other hand, if missing values are associated with specific segments of the population, the resulting analysis may be biased.

There are several choices to handle missing values (Tan, Steinbach, & Kumar, 2006):

a. *Ignore the missing value during analysis* – This approach ignores missing values in the analysis. This approach can, however, introduce inaccuracies in the results.
b. *Delete cases with missing values* – This approach eliminates the cases with missing values. It can however lead to bias in the analysis.
c. *Delete variables with lots of missing data* – This approach eliminates the variables with missing values. However, if those variables are deemed critical to the analysis, this approach many not be feasible.
d. *Estimate the missing values* – Sometimes missing values can be estimated. Common simple methods for estimation include mean and median substitution. Other more complex methods have been developed to estimate missing data. These methods are generally difficult to implement.

The Process of Data Cleaning

As a process, data cleaning consists of two main steps (Han & Kamber, 2006). The first step is *discrepancy detection*. Knowledge about the data or metadata is

crucial for this step. This knowledge could include the domain and data type of each variable, acceptable values, range of length of values, data dependencies, and so on. Summary statistics becomes very useful in understanding data trends and identifying any anomalies.

There are a number of automated tools that can aid in this step. They include *data scrubbing tools* which use simple domain knowledge to identify errors and make corrections in the data. They also include *data auditing tools* that find discrepancies by analyzing the data to discover rules and relationships, and identifying data that violates those rules and relationships.

The second step in data cleaning is *data transformation*, which defines and applies a series of transformations to correct the discrepancies identified in the first step. Data transformation is discussed in the next section. Automated tools can assist in the data transformation step. For example, *Data migration tools* allow simple transformations to be specified while *ETL (Extraction/Transformation/Loading) tools* allow users to specify data transformations using easy-to-use graphical user interface. Custom scripts need to be written for more complex transformations.

Data Transformation

Data transformation is the process of eliminating discrepancies and transforming the data into forms that are best suited for data mining (Osborne, 2002). Data transformation can involve the following operations:

a. *Data recoding* – Data recoding involves transforming the data type of a variable, for example, from alphanumeric to numeric.
b. *Smoothing* – Smoothing is used to remove noise from the data and reduce the effect of outliers. Smoothing techniques include binning, regression, and clustering.
c. *Grouping values* – Grouping values involves creating new variables with fewer categories by grouping together, or binning, values of existing variable.
d. *Aggregation* – Aggregation performs summary operations to the data.
e. *Generalization* – Generalization replaces low-level data with higher-level concepts.
f. *Functional transformation* – A functional transformation transforms distributions that skewed with a function, such as the natural log, to make it more regular and symmetrical.
g. *Normalization* – Normalization scales the values of a variable so that they fall with a small specified range, such as 0.0 to 1.0.
h. *Feature construction* – New fields are created and added to the data set to help the mining process.
i. *Feature selection* – Feature selection reduces the number of variables in the analysis for easier interpretation of the results.

FUTURE TRENDS

New approaches to data cleaning emphasize strong user involvement and interactivity. These approaches combine discrepancy detection and transformation into powerful automated tools (Raman & Hellertein, 2001). Using these tools, users build a series of transformations by creating individual transformations, one step at a time, using a graphical user interface. Results can be shown immediately on the data being manipulated on the screen. Users can elect to accept or undo the transformations. The tool can perform discrepancy checking automatically on the latest transformed view of the data. Users can develop and refine transformations as discrepancies are found, leading to a more effective and efficient data cleaning.

Another future trend for data cleaning is the development of declarative languages for the specification of data transformation operators (Galhardas, Florescu, Shasha, Simon, & Sita, 2001). These languages could be defined through powerful extensions to SQL and related algorithms to enable user to express data cleaning rules and specifications in an efficient manner.

CONCLUSION

Data preparation is a very important prerequisite to data mining as real-world data tend to suffer from inconsistencies, noise, errors and other problems. It is a broad area and consists of different approaches and techniques that are interrelated in complex ways. Data preparation can even answer some of the questions typically revealed by data mining. For example, patterns can sometimes be found, through visual inspection of data. Additionally, some of the techniques used in data

exploration can aid in understanding and interpreting data mining results.

REFERENCES

Berry, M., & Linoff, G. (2004). *Data mining techniques for marketing, sales, and customer relationship management* (2nd ed.). Indianapolis, IN: Wiley Publishing, Inc.

Bock, H. H. & Diday, E. (2000). *Analysis of symbolic data: Exploratory methods for extracting statistical information from complex data (Studies in classification, data analysis, and knowledge organizations)*. Berlin: Springer-Verlag.

Devore, J. L. (2003). *Probability and statistics for engineering and the sciences* (6th ed.). Duxbury Press.

Fayyad, U. M., Grinstein, G. G., & Wierse, A. (Eds). (2001). *Information visualization in data mining and knowledge discovery*. San Francisco, CA: Morgan Kaufmann Publishers.

Galhardas, H., Florescu, D., Shasha, D., Simon, E., & Sita, C.-A. (2001). Declarative data cleaning: Language, model, and algorithms. In *Proc. 2001 Int. Conf. on Very Large Data Bases (VLDB01)* (pp. 371–380).

Gaohua, F. H., & Liu, H. (2000). *Sampling and its application in data mining*: A survey (Tech. Rep. TRA6/00). Singapore: National University of Singapore.

Han, J., & Kamber, M. (2006). *Data mining: Concepts and techniques* (2nd ed.). San Francisco, CA: Morgan Kaufmann Publishers.

Liu, H., & Motoda, H. (1998). *Feature selection for knowledge discovery and data mining*. Kluwer Academic Publishers.

Olson, J. E. (2003). Data quality: *The accuracy dimension*. San Francisco, CA: Morgan Kaufmann Publishers.

Osborne, J. (2002). Notes on the use of data transformations. *Practical Assessment, Research and Evaluation*. *8*(6).

Ramakrishnan, R., & Gehrke, J. (2003). *Database management systems* (3rd ed.). McGraw-Hill.

Raman, V., & Hellertein, J. Ml (2001). Potter's wheel: An interactive data cleaning system. In *Proc.2001 Int. Conf. on Very Large Data Bases (VLDB01)* (pp. 381–390).

SPSS. (2002a). *Data mining: Data understanding and data preparation*. Chicago, IL: SPSS.

SPSS. (2002b). *Data Mining: Visualization, Reporting and Deployment*. Chicago, IL: SPSS.

Tan, P., Steinbach, M., & Kumar, V. (2006). *Introduction to data mining*. Boston, MA: Addison-Wesley.

Tukey, J. (1977). *Exploratory data analysis*. Boston, MA: Addison-Wesley.

Weiss, G. (2004). Mining with rarity: A unifying framework. *SIGKDD Explorations*. *6*(1), 7-19.

KEY TERMS

Data Cleaning: The detection and correction of data quality problems.

Data Exploration: The preliminary investigation of data in order to better understand its characteristics.

Data Preparation: The steps that should be applied to make data more suitable for data mining.

Data Reorganization: The process of reorganizing data into a different case basis.

Data Selection: The task of extracting smaller representative data sets from larger data sources.

Data Transformation: The process of transforming data into forms that are best suited for data mining.

Data Visualization: The display of data in tabular or graphic format.

Data Provenance

Vikram Sorathia
Dhirubhai Ambani Institute of Information and Communication Technology (DA-IICT), India

Anutosh Maitra
Dhirubhai Ambani Institute of Information and Communication Technology (DA-IICT), India

INTRODUCTION

In recent years, our sensing capability has increased manifold. The developments in sensor technology, telecommunication, computer networking and distributed computing domain have created strong grounds for building sensor networks that are now reaching global scales (Balazinska et al., 2007). As data sources are increasing, the task of processing and analysis has gone beyond the capabilities of conventional desktop data processing tools. For quite a long time, data was assumed to be available on the single user-desktop; and handling, processing as well as analysis was carried out single-handedly. With proliferation of streaming data-sources and near real-time applications, it has become important to make provisions of automated identification and attribution of data-sets derived from such diverse sources. Considering the sharing and reuse of such diverse data-sets, the information about: the source of data, ownership, time-stamps, accuracy related details, processes and transformations subjected to it etc. have become essential. The piece of data that provide such information about the given data-set is known as *Metadata*.

The need is recognized for creating and handling of metadata as an integrated part of large-scale systems. Considering the information requirements of scientific and research community, the efforts towards the building global data commons have came into existence (Onsrud & Campbell, 2007). A special type of service is required that can address the issues like: explication of licensing & Intellectual Property Rights, standards based automated generation of metadata, data provenance, archival and peer-review. While each of these terms is being addressed as individual research topics, the present article is focused only on Data Provenance.

BACKGROUND

In present scenario, data-sets are created, processed and shared at ever increasing quantities sometimes approaching gigabyte or terabyte scales. The given data-set is merely a useless series of characters, unless proper information is provided about it's content and quality. A good analogy provided to understand the importance of such metadata; a data-set can be compared with a piece of art. The art collectors and scholars are able to appreciate given object only based on authentic documented history that reveals the information like the artist, the era of creation, related events and any special treatments it was subjected to. Similarly, a given data-set may be a raw and uncorrected sensor log, or it can be derived after subjecting it to a series of careful statistical and analytical operations, making it useful for decision making.

Any kind of information provided about the data-set is therefore helpful in determining its true value for potential users. But a general record of information about given data-set; which also qualifies to be a Metadata, renders only a limited value to the user by providing simple housekeeping and ownership related information. It is only through a systematic documented history about source, intermediate stages and processing subjected to the given data-set captured in the metadata content, that provides due recognition of the quality and characteristic of the given data-set. This special type of metadata content is known as "Data Provenance" or "Pedigree" or "Lineage" or "Audit Trail". It is now being recognized (Jagadish et al., 2007) that apart from the data models, Data Provenance has critical role in improving usability of the data.

MAIN FOCUS

Provenance is used in art, science, technology and many other fields for a long time. With the recent developments

of database management systems (DBMS) responding to increasingly complex utilization scenarios, the importance of data provenance has become evident. In simplest from, data provenance in DBMS can be utilized to hold information about how, when any why the Views are created. Sensor networks, enterprise systems and collaborative applications involve more complex data provenance approaches required to handle large data stores (Ledlie, Ng, & Holland, 2005).

The purpose of this article is to introduce only the important terms and approaches involved in Data Provenance to the Data Mining and Data Warehousing community. For the detailed account of the subject including historical development, taxonomy of approaches, recent research and application trends, the readers are advised to refer (Bose & Frew, 2005) and (Simmhan, Plale, & Gannon, 2005).

Characteristics of Provenance

Data Provenance is a general umbrella term under which many flavors can be identified based on how it is created and what specific information it provides. In a simple way (Buneman, Khanna, & Tan, 2001) it is classified it in two terms: *Where Provenance* and *Why provenance*. Where provenance refers to the data that provides information about location at which the source of data can be found. Why provenance provide information about the reason due to which the data is in current state. Later, (Braun, Garfinkel, Holland, Muniswamy-Reddy, & Seltzer, 2006) identified more flavors of data provenance based on the handling approach. A manual entry of provenance information is identified as *Manual Provenance* or Annotation. The approach that relies on the observation by the system is called *Observed Provenance*. In case of observed provenance, the observing system may not be configured to record all possible intermediate stages that the given dataset may pass through. Hence, resulting provenance, holding only partial information due to incomplete information flow analysis is classified as *False Provenanc*e. When participating entities provide explicit provenance to the third-party provenance handling system, it is known as *Disclosed Provenance*. When a participating system directs the desired structure and content about the object, it is classified as *Specified Provenance*. Many other provenance characteristics are studied for the detail classification of data provenance in (Simmhan, Plale, & Gannon, 2005). It mentions the reference to "*Phantom Linage"*-a special status of provenance when the data-set that is being described in given provenance record is deleted from the source.

Provenance System Requirements

Comprehensive application scenario followed by detailed use cases analysis has been documented to reveal the requirements for building provenance systems (Miles, Groth, Branco, & Moreau, 2007). It is recommended that a system targeted to handle data provenance, must support some general features apart from any specific that can be identified based on the application domain. A core functional requirement is the activity of *Provenance Record* that requires the recording the provenance using manual or automated mechanism. The collected record must be stored and shared for the utilization - which leads to the requirement for a *Storage Service*. The recorded provenance is accessed by the end-users or systems by issuing appropriately formed queries. The support for query is an important feature that allows retrieval of provenance. Handling provenance is an event-driven methodology for tracking, executing and reporting series of execution steps or processes collected as a workflow. To support the execution of a workflow enactment service is required. A processing component is required that is responsible for creation and handling, validating and inference of provenance data. Any special Domain Specific information that is beyond the scope of general purpose provenance system, and must be provided with special provisions. *Actor side recording* is a functional requirement that allows recording of provenance data from actor side that cannot be observed or recorded by an automated system. Presentation functionality is required that can allow rendering of search-results in user-friendly manner that reveals the historical process hierarchy and source information.

Some Provenance Systems

Over a period of time, many provenance systems are proposed that realize one or more functionalities identified above. In Data Warehousing and Data Mining domain, it is conventional that the users themselves assume the responsibility for collecting, encoding and handling of the required data-sets from multiple sources and maintenance the provenance information. Such manually curated data provenance poses a challenge

due to heterogeneity in collaborative environment. A "copy-paste database" approach (Buneman, Chapman, Cheney, & Vansummeren, 2006) is proposed as a solution to this problem that emulates a human curator adding individual items in the provenance. A formal treatment towards building a data provenance system, covering tasks from modeling to implementation and storage is discussed in (Lawabni, Hong, Du, & Tewfik, 2005). Considering the information needs of users in bio-informatics and remote sensing data sources, an Update Propagation Module is proposed focusing on Sensitivity analysis, Confidence level and Complexity in handling provenance data. Considering the interoperability issues in handling multiple heterogeneous systems, various designs are proposed that provides implementation of grid (Wootten, Rajbhandari, Rana, & Pahwa, 2006), semantics (Frey et al., 2006) and workflow (Barga & Digiampietri, 2006) based provenance systems.

Automation in Provenance

Apart from manual and observed provenance, it is possible to automate the process of provenance handling if some automation related issues are duly addressed in the design. (Braun, Garfinkel, Holland, Muniswamy-Reddy, & Seltzer, 2006) pointed out issues like, predicting granularity of observed provenance to match the user needs, the possibility of cyclic ancestry and determination of a new version based on a unit activity. In case of workflow, the workflow enactment engine is made responsible for automated handling of provenance (Barga & Digiampietri, 2006). A multi-layered model employed for handling different levels of workflow details in representation, thereby allowing multiple granularity of data provenance to suit the user needs.

Provenance in SOA and Grid

Service Oriented Architecture (SOA) paradigm advocates the strategy to expose unit functionality as a standard based service. (Tsai et al., 2007) provides special challenges for creating and handling of Data Provenance following SOA paradigm, with identification of additional features like dynamic composition, data routing and data classification. It is further claimed that complete recording of the provenance is not required and based on the needs, in place of full provenance record, it can be restricted only as: actor provenance, time-based provenance, event-based provenance, actor related provenance and scenario-based provenance. In SOA domain, Grid Computing is a special branch focused on ensuring non-trivial quality of service in providing access to services and resources that can be used for both storing and processing data. Utility of Grid in e-science projects have been successfully demonstrated by many projects. Data provenance in Grid environment poses unique requirements that are identified (Groth, Luck, & Moreau, 2004) by introduction of SOA specific flavors of the data provenance. The information about service invocations and interactions that resulted the data to its current form is introduced as *Interaction Provenance*.

It is likely that certain experiments have data processing needs that cannot be met by single service invocation. Therefore more than one service must be integrated to create a workflow that realizes the complete process. Such experiment-specific information about the process that can only be provided by a specific actor, and that can not be verifiable by any other means is identified as *Actor Provenance*. Up on building such composition, a service-based workflow encompassing a complete multi-stage business or scientific process can be achieved. (Szomszor & Moreau, 2003) provides a provenance architecture that offers a Provenance Service that exposes record and query operation in Grid Environment. During the execution of the workflow, for each unit service, the record operation is invoked. This approach results in a workflow trace, that consists of the provenance detail of each invocation during the workflow.

Provenance in Applications

The field of bio-informatics is claimed to be a field of Knowledge Management, where data from various sources are passed through series of processes resulting in important bio-informatics resource. Therefore, in this domain, the activity of knowledge management for *In Silico* experiments is considered as a task of managing Data Provenance. Addressing this challenge (Stevens, Zhao, & Goble, 2007) proposed an approach identifying four levels or data provenance information namely: Process level, data level, organization level and knowledge level in a provenance pyramid. Provenance handling using such approach provided capability for verification and validation of experiments.

In field of astronomy a virtual observatory effort provides access to established astronomy data to multiple users. Astronomy Distributed Annotation System was demonstrated (Bose, Mann, & Prina-Ricotti, 2006) that allows assertion of entity mappings that can be shared and integrated with existing query mechanisms for better performance in distributed environment.

Recently, wide acceptance of Web 2.0 application has resulted in proliferation of user-created data contributed by numerous users acting as data sources. When such data-sources are utilized for the determination of ranking and evaluation of products and services, authenticity of the content becomes important challenge. (Golbeck, 2006) proposed an approach that captures trust annotation and provenance for inference of trust relationships. It demonstrated a case for calculating personalized movie recommendations and ordering reviews based on computed trust values. Other interesting case based on the same principle, demonstrated how intelligence professionals assign the trust value to the provider of information and analysis to ascertain the quality of the information content.

In Healthcare domain, management of the patient data is a challenging task as such data is available in the form of islands of information created by one or many health professionals (Kifor et al., 2006). Agent based technology was found useful in efforts towards the automation, but face challenges due to heterogeneity of systems and models employed by individual professionals. As a solution to this problem, a Provenance based approach is proposed that requires the professional to assert the treatment related information encoded in process calculus and semantics. A case of organ transplant management application demonstrated how multiple agents collaborate for handling the data provenance so that, consistent health records can be generated.

Apart from social applications and project specific research problems, recent efforts of grid computing research community have contributed general-purpose systems that provide higher computational, storage and processing capability exposed through open general purpose standards. The resulting grid technology is found ideal for execution and handling of large-scale e-science experiments in distributed environments. While, by virtue of the capabilities of such systems, computation and resource management for generating experimental processes became considerably easy task, it introduced a novel requirement for evaluation of completed experiments for their validity. A provenance based approach is demonstrated (Miles, Wong, et al., 2007), that enables the verification of documented scientific experiment, by evaluating: input arguments to the scientific processes, the source of data, verification against Ontology, conformance to the plans and patent related details.

FUTURE TRENDS

As data sources continues to proliferate, data sharing and processing capabilities continue to grow and as the end-users continue to utilize them, the issue of data provenance will continue to become more complex. The piling up of the raw data and multiple copies or derived data-sets for each of them, calls for investigation of optimal engineering approaches that can provide low-cost, standards-based, reliable handling for error-free data provenance. Apart from automation achieved using semantics or workflow based strategy, the future research efforts will focus on machine learning techniques and similar approaches that will enable machine processable handling, storage, dissemination and purging of the data provenance.

CONCLUSION

As monitoring capabilities are increasing, it is become possible to capture, handle and share large amounts of data. The data collected initially for specific purpose can later be useful to many other users in their decision-making process. From initial collection phase up to the phase in which data is used for decision-making, data undergoes various transformations. It therefore becomes a challenging problem to identify and determine true identity and quality of given data-set extracted out of large amount of data shared in huge repositories. Data Provenance- a special type of metadata, is introduced as a solution for this problem. With increasing acceptance of service-oriented approaches for data sharing and processing, the resulting loosely coupled systems provide very little support for consistent management of provenance. Once published data is subjected to unplanned used and further processed by many intermediate users, the maintenance of change-log of the data becomes a challenging issue. This article provided various methods, protocols and system architectures

relating to data provenance in specific domain and application scenarios.

REFERENCES

Balazinska, M., Deshpande, A., Franklin, M. J., Gibbons, P. B., Gray, J., Hansen, M., et al. (2007). Data management in the worldwide sensor web. *IEEE Pervasive Computing*, *6*(2), 30–40.

Barga, R. S., & Digiampietri, L. A. (2006). Automatic generation of workflow provenance. In L. Moreau & I. T. Foster (Eds.), *IPAW* (Vol. 4145, p. 1-9). Springer.

Bose, R., & Frew, J. (2005). Lineage retrieval for scientific data processing: a survey. *ACM Comput.Surv.*, *37*(1), 1–28.

Bose, R., Mann, R. G., & Prina-Ricotti, D. (2006). Astrodas: Sharing assertions across astronomy catalogues through distributed annotation. In L. Moreau & I. T. Foster (Eds.), *IPAW* (Vol. 4145, p. 193-202). Springer.

Braun, U., Garfinkel, S. L., Holland, D. A., Muniswamy-Reddy, K.-K., & Seltzer, M. I. (2006). Issues in automatic provenance collection. In L. Moreau & I. T. Foster (Eds.), *IPAW* (Vol. 4145, p. 171-183). Springer.

Buneman, P., Chapman, A., Cheney, J., & Vansummeren, S. (2006). A provenance model for manually curated data. In L. Moreau & I. T. Foster (Eds.), *IPAW* (Vol. 4145, p. 162-170). Springer.

Buneman, P., Khanna, S., & Tan, W. C. (2001). Why and where: A characterization of data provenance. *ICDT '01: Proceedings of the 8th International Conference on Database Theory* (pp. 316–330). London, UK: Springer-Verlag.

Frey, J. G., Roure, D. D., Taylor, K. R., Essex, J.W., Mills, H. R., & Zaluska, E. (2006). Combechem: A case study in provenance and annotation using the semantic web. In L. Moreau & I. T. Foster (Eds.), *IPAW* (Vol. 4145, p. 270-277). Springer.

Golbeck, J. (2006). Combining provenance with trust in social networks for semantic web content filtering. In L. Moreau & I. T. Foster (Eds.), *IPAW* (Vol. 4145, p. 101-108). Springer.

Groth, P. T., Luck, M., & Moreau, L. (2004). A protocol for recording provenance in service-oriented grids. *OPODIS* (p. 124-139).

Jagadish, H. V., Chapman, A., Elkiss, A., Jayapandian, M., Li, Y., Nandi, A., et al. (2007). Making database systems usable. *SIGMOD '07: Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data* (pp. 13–24). New York: ACM Press.

Kifor, T., Varga, L. Z., Vazquez-Salceda, J., Alvarez, S., Willmott, S., Miles, S., et al. (2006). Provenance in agent-mediated healthcare systems. *IEEE Intelligent Systems*, *21*(6), 38–46.

Lawabni, A. E., Hong, C., Du, D. H. C., & Tewfik, A. H. (2005). A novel update propagation module for the data provenance problem: A contemplating vision on realizing data provenance from models to storage. *MSST '05: Proceedings of the 22nd IEEE / 13th NASA Goddard Conference on Mass Storage Systems and Technologies* (pp. 61–69). Washington, DC: IEEE Computer Society.

Ledlie, J., Ng, C., & Holland, D. A. (2005). Provenance-aware sensor data storage. *ICDEW '05: Proceedings of the 21st International Conference on Data Engineering Workshops*, (p. 1189). Washington, DC: IEEE Computer Society.

Miles, S., Groth, P., Branco, M., & Moreau, L. (2007). The requirements of using provenance in e-science experiments. *Journal of Grid Computing*, *5*(1), 1-25.

Miles, S., Wong, S. C., Fang, W., Groth, P., Zauner, K.-P., & Moreau, L. (2007). Provenance-based validation of e-science experiments. *Web Semantics*, *5*(1), 28–38.

Onsrud, H., & Campbell, J. (2007). Big opportunities in access to "small science" data. *Data Science Journal. Special Issue: Open Data for Global Science*, *6*, 58-66.

Simmhan, Y. L., Plale, B., & Gannon, D. (2005). A survey of data provenance in e-science. *SIGMOD Rec.*, *34*(3), 31–36.

Stevens, R., Zhao, J., & Goble, C. (2007). Using provenance to manage knowledge of in silico experiments. *Brief Bioinform.*, *8*(3), 183-194.

Szomszor, M., & Moreau, L. (2003). Recording and reasoning over data provenance in web and grid services. In *Coopis/doa/odbase* (p. 603-620).

Tsai, W.-T., Wei, X., Zhang, D., Paul, R., Chen, Y., & Chung, J.-Y. (2007). A new SOA data provenance framework. ISADS '07: *Proceedings of the Eighth International Symposium on Autonomous Decentralized Systems* (pp. 105–112). Washington, DC: IEEE Computer Society.

Wootten, I., Rajbhandari, S., Rana, O., & Pahwa, J. S. (2006). Actor provenance capture with ganglia. *CCGrid '06: Proceedings of the Sixth IEEE International Symposium on Cluster Computing and the Grid* (pp. 99–106). Washington, DC: IEEE Computer Society.

KEY TERMS

Actor Provenance: A special provenance strategy utilized in SOA realizing a scientific workflow for which a domain specific information can be collected only by the actors and by no other automated mechanisms.

Data Provenance: Data provenance is a historical record of processes executed over the raw data that resulted in to its current form. It is a special kind of Metadata that is also known as "Lineage", "Audit trail" or "Pedigree".

Interaction Provenance: A Provenance Strategy utilized in SOA environment that captures every method call in data provenance.

Phantom Linage: A special kind of provenance data that describes the data that is missing or deleted from the sources.

Provenance Architecture: A system architecture that is capable to satisfy all the functional and non-functional requirements identified for a provenance in given problem.

Provenance Collection: Activity of collecting provenance from single or multiple data sources and processing systems by employing manual or automated methods.

Provenance Granularity: The detail of provenance information collected and recorded. The high granularity contains every small intermediate process step (or method call) where as coarse granularity provenance provides only major changes experienced by the dataset.

Observed Provenance: Presentation of data in human understandable graphics, images, or animation.

Data Quality in Data Warehouses

William E. Winkler
U.S. Bureau of the Census, USA

INTRODUCTION

Fayyad and Uthursamy (2002) have stated that the majority of the work (representing months or years) in creating a data warehouse is in cleaning up duplicates and resolving other anomalies. This paper provides an overview of two methods for improving quality. The first is record linkage for finding duplicates within files or across files. The second is edit/imputation for maintaining business rules and for filling-in missing data. The fastest record linkage methods are suitable for files with hundreds of millions of records (Winkler, 2004a, 2008). The fastest edit/imputation methods are suitable for files with millions of records (Winkler, 2004b, 2007a).

BACKGROUND

When data from several sources are successfully combined in a data warehouse, many new analyses can be done that might not be done on individual files. If duplicates are present within a file or across a set of files, then the duplicates might be identified. Record linkage uses name, address and other information such as income ranges, type of industry, and medical treatment category to determine whether two or more records should be associated with the same entity. Related types of files might be combined. In the health area, a file of medical treatments and related information might be combined with a national death index. Sets of files from medical centers and health organization might be combined over a period of years to evaluate the health of individuals and discover new effects of different types of treatments. Linking files is an alternative to exceptionally expensive follow-up studies.

The uses of the data are affected by *lack of quality* due to duplication of records and missing or erroneous values of variables. Duplication can waste money and yield error. If a hospital has a patient incorrectly represented in two different accounts, then the hospital might repeatedly bill the patient. Duplicate records may inflate the numbers and amounts in overdue billing categories. If the quantitative amounts associated with some accounts are missing, then the totals may be biased low. If values associated with variables such as billing amounts are erroneous because they do not satisfy edit or business rules, then totals may be biased low or high. Imputation rules can supply replacement values for erroneous or missing values that are consistent with the edit rules and preserve joint probability distributions. Files without error can be effectively data mined.

MAIN THRUST OF THE CHAPTER

This section provides an overview of record linkage and of statistical data editing and imputation. The cleaning-up and homogenizing of the files are preprocessing steps prior to data mining.

Record Linkage

Record linkage is also referred to as *entity resolution* or *object identification*. Record linkage was given a formal mathematical framework by Fellegi and Sunter (1969). Notation is needed. Two files A and B are matched. The idea is to classify pairs in a product space **A ´ B** from two files A and B into M, the set of true matches, and U, the set of true nonmatches. Fellegi and Sunter considered ratios of conditional probabilities of the form:

$$R = P(\gamma \in \Gamma \mid M) / P(\gamma \in \Gamma \mid U) \quad (1)$$

where γ is an arbitrary agreement pattern in a comparison space Γ over all the pairs in **A ´ B**. For instance, Γ might consist of eight patterns representing simple agreement or not on the largest name component, street name, and street number. The distinct patterns in Γ partition the entire set of pairs in **A ´ B**. Alternatively, each $\gamma \in \Gamma$ might additionally account for the relative frequency with which specific values of name components such as “Smith”, “Zabrinsky”, “AAA”, and “Capitol” oc-

cur. Ratio R or any monotonely increasing function of it such as the natural log is referred to as a *matching weight (or score).*

The decision rule is given by:

If $R > T_{\mu}$, then designate pair as a match.

If $T_{\lambda} \leq R \leq T_{\mu}$, then designate pair as a possible match and hold for clerical review. (2)

If $R < T_{\lambda}$, then designate pair as a nonmatch.

The cutoff thresholds T_{μ} and T_{λ} are determined by a priori error bounds on false matches and false nonmatches. Rule (2) agrees with intuition. If $\gamma \in \Gamma$ consists primarily of agreements, then it is intuitive that $\gamma \in \Gamma$ would be more likely to occur among matches than nonmatches and ratio (1) would be large. On the other hand, if $\gamma \in \Gamma$ consists primarily of disagreements, then ratio (1) would be small. Rule (2) partitions the set $\gamma \in \Gamma$ into three disjoint subregions. The region $T_{\lambda} \leq R \leq T_{\mu}$ is referred to as the no-decision region or clerical review region. In some situations, resources are available to review pairs clerically.

Linkages can be error-prone in the absence of *unique identifiers* such as a verified social security number that identifies an individual record or entity. *Quasi identifiers* such as name, address and other non-uniquely identifying information are used. The combination of quasi identifiers can determine whether a pair of records represents the same entity. If there are errors or differences in the representations of names and addresses, then many duplicates can erroneously be added to a warehouse. For instance, a business may have its name 'John K Smith and Company' in one file and 'J K Smith, Inc' in another file. Without additional corroborating information such as address, it is difficult to determine whether the two names correspond to the same entity. With three addresses such as '123 E. Main Street,' '123 East Main St' and 'P O Box 456' and the two names, the linkage can still be quite difficult. With suitable pre-processing methods, it may be possible to represent the names in forms in which the different components can be compared. To use addresses of the forms '123 E. Main Street' and 'P O Box 456,' it may be necessary to use an auxiliary file or expensive follow-up that indicates that the addresses have at some time been associated with the same entity.

If there is minor typographical error in individual fields, then string comparators that account for typographical error can allow effective comparisons (Winkler, 2004b; Cohen, Ravikumar, & Fienberg, 2003). Individual fields might be first name, last name, and street name that are delineated by extraction and standardization software. Rule-based methods of standardization are available in commercial software for addresses and in other software for names (Winkler, 2008). The probabilities in equations (1) and (2) are referred to as *matching parameters*. If training data consisting of matched and unmatched pairs is available, then a *supervised method* that requires training data can be used for estimation matching parameters. Optimal matching parameters can sometimes be estimated via unsupervised learning methods such as the EM algorithm under a conditional independence assumption (also known as *naïve Bayes* in machine learning). The parameters vary significantly across files (Winkler, 2008). They can even vary significantly within a single file for subsets representing an urban area and an adjacent suburban area. If two files each contain 10,000 or more records, than it is impractical to bring together all pairs from two files. This is because of the small number of potential matches within the total set of pairs. *Blocking* is the method of only considering pairs that agree exactly (character-by-character) on subsets of fields. For instance, a set of blocking criteria may be to only consider pairs that agree on U.S. Postal ZIP code and first character of the last name. Additional blocking passes may be needed to obtain matching pairs that are missed by earlier blocking passes (Winkler, 2004a).

Statistical Data Editing and Imputation

Correcting inconsistent information and filling-in missing information needs to be efficient and cost-effective. For single fields, edits are straightforward. A look-up table may show that a given value in not in an acceptable set of values. For multiple fields, an edit might require that an individual of less than 15 years of age must have marital status of unmarried. If a record fails this edit, then a subsequent procedure would need to change either the age or the marital status. Alternatively, an edit might require than the ratio of the total payroll divided by the number of employees at a company within a particular industrial

classification fall within lower and upper bounds. In each situation, the 'errors' in values in fields may be due to typographical error or misunderstanding of the information that was requested.

Editing has been done extensively in statistical agencies since the 1950s. Early work was clerical. Later computer programs applied if-then-else rules with logic similar to the clerical review. The main disadvantage was that edits that did not fail for a record initially would fail as values in fields associated with edit failures were changed. Fellegi and Holt (1976, hereafter FH) provided a theoretical model. In providing their model, they had three goals:

1. The data in each record should be made to satisfy all edits by changing the fewest possible variables (fields).
2. Imputation rules should derive automatically from edit rules.
3. When imputation is necessary, it should maintain the joint distribution of variables.

FH (Theorem 1) proved that implicit edits are needed for solving the problem of goal 1. Implicit edits are those that can be logically derived from explicitly defined edits. Implicit edits provide information about edits that do not fail initially for a record but may fail as values in fields associated with failing edits are changed. The following example illustrates some of the computational issues. An edit can be considered as a set of points. Let edit E = {married & age ≤ 15}. Let *r* be a data record. Then $r \in E \Rightarrow r$ fails edit. This formulation is equivalent to 'If age ≤15, then not married.' If a record *r* fails a set of edits, then one field in each of the failing edits must be changed. An implicit edit E_3 can be implied from two explicitly defined edits E_1 and E_2; i.e., E_1 & $E_2 \Rightarrow E_3$.

E_1 = {age ≤ 15, married, . }
E_2 = { . , not married, spouse}
E_3 = {age ≤ 15, . , spouse}

The edits restrict the fields *age*, *marital-status* and *relationship-to-head-of- household*. Implicit edit E_3 is derived from E_1 and E_2. If E_3 fails for a record *r* = {age ≤ 15, not married, spouse}, then necessarily either E_1 or E_2 fail. Assume that the implicit edit E_3 is unobserved. If edit E_2 fails for record *r*, then one possible correction is to change the marital status field in record *r* to married to obtain a new record r_1. Record r_1 does not fail for E_2 but now fails for E_1. The additional information from edit E_3 assures that record *r* satisfies all edits after changing one additional field. For larger data situations having more edits and more fields, the number of possibilities increases at a very high exponential rate.

In data-warehouse situations, the ease of implementing FH ideas using generalized edit software is dramatic. An analyst who has knowledge of the edit situations might put together the edit tables in a relatively short time. In a direct comparison, Statistics Canada and the U.S. Census Bureau (Herzog, Scheuren & Winkler, 2007) compared two edit systems on the same sets of economic data. Both were installed and run in less than one day. Both were able to produce data files that were as good as the manually edited files that were used as a reference. For many business and data warehouse situations, only a few simple edit rules might be needed. In a dramatic demonstration, researchers at the Census Bureau (Herzog et al., 2007) showed that a FH system was able to edit/impute a large file of financial information in less than 24 hours. For the same data, twelve analysts needed six months and changed three times as many fields.

FUTURE TRENDS

Various methods represent the most recent advances in areas that still need further extensions to improve speed significantly, make methods applicable in a variety of data types or reduce error. First, some research considers better search strategies to compensate for typographical error. Winkler (2008, 2004a) applies efficient blocking strategies for bringing together pairs except in the extreme situations where some truly matching pairs have no 3-grams in common (i.e., no consecutive three characters in one record are in its matching record). Although extremely fast, the methods are not sufficiently fast for the largest situations (larger than 10^{17} pairs). Second, another research area investigates methods to standardize and parse general names and address fields into components that can be more easily compared. Cohen and Sarawagi (2004) and Agichtein and Ganti (2004) have Hidden Markov methods that work as well or better than some of the rule-based methods. Although the Hidden Markov methods require training data, straightforward methods quickly create

additional training data for new applications. Third, in the typical situations with no training data, optimal matching parameters are needed for separating matches and nonmatches even in the case when error rates are not estimated. Ravikumar and Cohen (2004) have unsupervised learning methods that improve over EM methods under the conditional independence assumption. With some types of data, their methods are even competitive with supervised learning methods. Bhattacharya and Getoor (2006) introduced unsupervised learning techniques based on latent Dirichlet models that may be more suitable than predecessor unsupervised methods. Fourth, other research considers methods of (nearly) automatic error rate estimation with little or no training data. Winkler (2002) considers semi-supervised methods that use *unlabeled* data and very small subsets of *labeled* data (training data). Winkler (2006) provides methods for estimating false match rates without training data. The methods are primarily useful in situations with matches and nonmatches can be separated reasonably well. Fifth, some research considers methods for adjusting statistical and other types of analyses for matching error. Lahiri and Larsen (2004) have methods for improving the accuracy of statistical analyses in the presence of matching error in a very narrow range of situations. The methods need to be extended to situations with moderate false match rates (0.10+), with more than two files, with sets of matched pairs that are unrepresentative of the pairs in one or more of the files being matches, and with complex relationships of the variables.

There are two trends for edit/imputation research. The first is faster ways of determining the minimum number of variables containing values that contradict the edit rules. Using satisfiability, Bruni (2005) provides very fast methods for editing discrete data. Riera-Ledesma and Salazar-Gonzalez (2007) apply clever heuristics in the set-up of the problems that allow direct integer programming methods for continuous data to perform far faster. Although these methods are considerably faster than predecessor methods with large test decks (of sometimes simulated data), the methods need further testing with large, real-world data sets. The second is methods that preserve both statistical distributions and edit constraints in the pre-processed data. Winkler (2003) connects the generalized imputation methods of Little and Rubin (2002) with generalized edit methods. For arbitrary discrete data and restraint relationships, Winkler (2007a) provides fast methods of edit/imputation that generalize the methods of imputation based on Bayesian networks introduced by Di Zio, Scanu, Coppola, Luzi and Ponti (2004) and the statistical matching methods of D'Orazio, Di Zio and Scanu (2006). The methods use convex constraints that cause the overall data to better conform to external and other controls. Statistical matching is sometimes used in situations where there are insufficient quasi-identifying fields for record linkage. As a special case (Winkler, 2007b), the methods can be used for generating synthetic or partially synthetic data that preserve analytic properties while significantly reducing the risk of re-identification and assuring the privacy of information associated with individuals. These methods assure, in a principled manner, the statistical properties needed for privacy-preserving data mining. Methods for general data (discrete and continuous) need to be developed.

CONCLUSION

To data mine effectively, data need to be pre-processed in a variety of steps that remove duplicates, perform statistical data editing and imputation, and do other clean-up and regularization of the data. If there are moderate errors in the data, data mining may waste computational and analytic resources with little gain in knowledge.

REFERENCES

Agichtein, E., & Ganti, V. (2004). Mining reference tables for automatic text segmentation, *ACM SIGKDD 2004*, 20-29.

Bhattacharya, I., & Getoor, L. (2006). A latent dirichlet allocation model for entity resolution. *The 6th SIAM Conference on Data Mining*.

Bruni, R. (2005). Error correction in massive data sets. *Optimization Methods and Software*, 20 (2-3), 295-314.

Cohen, W. W., & Sarawagi, S. (2004). Exploiting dictionaries in named entity extraction: combining semi-markov extraction processes and data integration methods. *ACM SIGKDD 2004*, 89-98.

D'Orazio, M., Di Zio, M., & Scanu, M. (2006). Statistical matching for categorical data: displaying uncertainty and using logical constraints. *Journal of Official Statistics*, 22 (1), 137-157.

Di Zio, M., Scanu, M., Coppola, L., Luzi, O., & Ponti, A. (2004). Bayesian networks for imputation. *Journal of the Royal Statistical Society, A*, 167(2), 309-322.

Fayyad, U., & Uthurusamy, R. (2002). Evolving data mining into solutions for insights. *Communications of the Association of Computing Machinery*, 45(8), 28-31.

Fellegi, I. P., & Holt, D. (1976). A systematic approach to automatic edit and imputation. *Journal of the American Statistical Association, 71*, 17-35.

Fellegi, I. P., & Sunter, A. B. (1969). A theory of record linkage. *Journal of the American Statistical Association*, 64, 1183-1210.

Herzog, T. A., Scheuren, F., & Winkler, W. E. (2007). *Data quality and record linkage techniques*. New York, N.Y.: Springer.

Lahiri, P. A., & Larsen, M. D. (2004). Regression analysis with linked data. *Journal of the American Statistical Association*, 100, 222-230.

Little, R. A., & Rubin, D. B., (2002). *Statistical analysis with missing data (2nd edition)*. New York: John Wiley.

Ravikumar, P., & Cohen, W. W. (2004). A hierarchical graphical model for record linkage. *Proceedings of the Conference on Uncertainty in Artificial Intelligence 2004*, Banff, Calgary, CA, July 2004. URL http://www.cs.cmu.edu/~wcohen

Riera-Ledesma, J., & Salazar-Gonzalez, J.-J. (2007). A branch-and-cut algorithm for the error localization problem in data cleaning. *Computers and Operations Research*. 34 (9), 2790-2804.

Winkler, W. E. (2002). Record linkage and Bayesian networks. *Proceedings of the Section on Survey Research Methods, American Statistical Association*, CD-ROM. URL http://www.census.gov/srd/papers/pdf/rrs2002-05.pdf.

Winkler, W. E. (2003). A contingency table model for imputing data satisfying analytic constraints. *American Statistical Association, Proc. Survey Research Methods Section*, CD-ROM. URL http://www.census.gov/srd/papers/pdf/rrs2003-07.pdf.

Winkler, W. E. (2004a). Approximate string comparator search strategies for very large administrative lists. *Proceedings of the Section on Survey Research Methods, American Statistical Association*, CD-ROM. URL http://www.census.gov/srd/papers/pdf/rrs2005-02.pdf.

Winkler, W. E. (2004b). Methods for evaluating and creating data quality. *Information Systems*, 29 (7), 531-550.

Winkler, W. E. (2006). Automatically estimating record linkage false match rates. *Proceedings of the Section on Survey Research Methods, American Statistical Association*. CD-ROM. URL http://www.census.gov/srd/papers/pdf/rrs2007-05.pdf.

Winkler, W. E. (2007a). General methods and algorithms for modeling and imputing discrete data under a variety of constraints. To appear at http://www.census.gov/srd/www/byyear.html.

Winkler, W. E. (2007b). Analytically valid discrete microdata files and re-identification. *Proceedings of the Section on Survey Research Methods, American Statistical Association*, to appear. Also at URL http://www.census.gov/srd/www/byyear.html.

Winkler, W. E. (2008). Record linkage. In C. R. Rao & D. Pfeffermann (eds.), *Sample Surveys: Theory, Methods and Inference*. New York, N. Y.: North-Holland.

KEY TERMS

Edit Restraints: Logical restraints such as business rules that assure that an employee's listed salary in a job category is not too high or too low or that certain contradictory conditions such as a male hysterectomy do not occur.

Imputation: The method of filling in missing data that sometimes preserves statistical distributions and satisfies edit restraints.

Quasi Identifiers: Fields such as name, address, and date-of-birth that by themselves do not uniquely identify an individual but in combination may uniquely identify.

Pre-Processed Data: In preparation for data mining, data that have been through consistent coding of fields, record linkage or edit/imputation.

Privacy-Preserving Data Mining: Mining files in which obvious identifiers have been removed and combinations of quasi-identifiers have been altered to both assure privacy for individuals and still maintain certain aggregate data characteristics for mining and statistical algorithms.

Record Linkage: The methodology of identifying duplicates in a single file or across a set of files using name, address, and other information.

Rule Induction: Process of learning, from cases or instances, if-then rule relationships consisting of an antecedent (if-part, defining the preconditions or coverage of the rule) and a consequent (then-part, stating a classification, prediction, or other expression of a property that holds for cases defined in the antecedent).

Training Data: A representative subset of records for which the truth of classifications and relationships is known and that can be used for rule induction in machine learning models.

Data Reduction with Rough Sets

Richard Jensen
Aberystwyth University, UK

Qiang Shen
Aberystwyth University, UK

INTRODUCTION

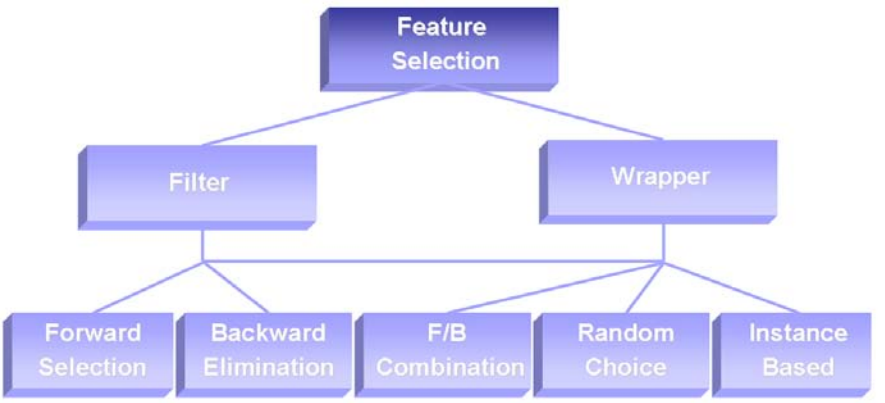
Data reduction is an important step in knowledge discovery from data. The high dimensionality of databases can be reduced using suitable techniques, depending on the requirements of the data mining processes. These techniques fall in to one of the following categories: those that transform the underlying meaning of the data features and those that are semantics-preserving. Feature selection (FS) methods belong to the latter category, where a smaller set of the original features is chosen based on a subset evaluation function. The process aims to determine a minimal feature subset from a problem domain while retaining a suitably high accuracy in representing the original features. In knowledge discovery, feature selection methods are particularly desirable as they facilitate the interpretability of the resulting knowledge. For this, rough set theory has been successfully used as a tool that enables the discovery of data dependencies and the reduction of the number of features contained in a dataset using the data alone, while requiring no additional information.

BACKGROUND

The main aim of feature selection is to determine a minimal feature subset from a problem domain while retaining a suitably high accuracy in representing the original features. In many real world problems FS is a must due to the abundance of noisy, irrelevant or misleading features. A detailed review of feature selection techniques devised for classification tasks can be found in (Dash & Liu, 1997).

The usefulness of a feature or feature subset is determined by both its relevancy and its redundancy. A feature is said to be relevant if it is predictive of the decision feature(s) (i.e. dependent variable(s)), otherwise it is irrelevant. A feature is considered to be redundant if it is highly correlated with other features. Hence, the

Figure 1. Feature selection taxonomy



search for a good feature subset involves finding those features that are highly correlated with the decision feature(s), but are uncorrelated with each other.

A taxonomy of feature selection approaches can be seen in Figure 1. Given a feature set of size n, the task of any FS method can be seen as a search for an ''optimal'' feature subset through the competing 2^n candidate subsets. The definition of what an optimal subset is may vary depending on the problem to be solved. Although an exhaustive method may be used for this purpose in theory, this is quite impractical for most datasets. Usually FS algorithms involve heuristic or random search strategies in an attempt to avoid this prohibitive complexity.

Determining subset optimality is a challenging problem. There is always a trade-off in non-exhaustive techniques between subset minimality and subset suitability - the task is to decide which of these must suffer in order to benefit the other. For some domains (particularly where it is costly or impractical to monitor many features, such as complex systems monitoring (Shen & Jensen, 2004)), it is much more desirable to have a smaller, less accurate feature subset. In other areas it may be the case that the modeling accuracy (e.g. the classification rate) using the selected features must be extremely high, at the expense of a non-minimal set of features, such as web content categorization (Jensen & Shen, 2004b).

MAIN FOCUS

The work on rough set theory offers an alternative, and formal, methodology that can be employed to reduce the dimensionality of datasets, as a preprocessing step to assist any chosen method for learning from data. It helps select the most information rich features in a dataset, without transforming the data, while attempting to minimize information loss during the selection process. Computationally, the approach is highly efficient, relying on simple set operations, which makes it suitable as a preprocessor for techniques that are much more complex. Unlike statistical correlation reducing approaches, it requires no human input or intervention. Most importantly, it also retains the semantics of the data, which makes the resulting models more transparent to human scrutiny. Combined with an automated intelligent modeler, say a fuzzy system or a neural network, the feature selection approach based on rough set theory can not only retain the descriptive power of the learned models, but also allow simpler system structures to reach the knowledge engineer and field operator. This helps enhance the interoperability and understandability of the resultant models and their reasoning.

Rough Set Theory

Rough set theory (RST) has been used as a tool to discover data dependencies and to reduce the number of attributes contained in a dataset using the data alone, requiring no additional information (Pawlak, 1991; Polkowski, 2002; Skowron et al., 2002). Over the past ten years, RST has become a topic of great interest to researchers and has been applied to many domains. Indeed, since its invention, this theory has been successfully utilized to devise mathematically sound and often, computationally efficient techniques for addressing problems such as hidden pattern discovery from data, data reduction, data significance evaluation, decision rule generation, and data-driven inference interpretation (Pawlak, 2003). Given a dataset with discretized attribute values, it is possible to find a subset (termed a *reduct*) of the original attributes using RST that are the most informative; all other attributes can be removed from the dataset with minimal information loss.

The rough set itself is the approximation of a vague concept (set) by a pair of precise concepts, called lower and upper approximations, which are a classification of the domain of interest into disjoint categories. The lower approximation is a description of the domain objects which are known with certainty to belong to the subset of interest, whereas the upper approximation is a description of the objects which possibly belong to the subset. The approximations are constructed with regard to a particular subset of features.

Rough set theory possesses many features in common (to a certain extent) with the Dempster-Shafer theory of evidence (Skowron & Grzymala-Busse, 1994) and fuzzy set theory (Wygralak, 1989). It works by making use of the granularity structure of the data only. This is a major difference when compared with Dempster-Shafer theory and fuzzy set theory, which require probability assignments and membership values, respectively. However, this does not mean that no model assumptions are made. In fact, by using only the given

information, the theory assumes that the data is a true and accurate reflection of the real world (which may not be the case). The numerical and other contextual aspects of the data are ignored which may seem to be a significant omission, but keeps model assumptions to a minimum.

Dependency Function-Based Reduction

By considering the union of the lower approximations of all concepts in a dataset with respect to a feature subset, a measure of the quality of the subset can be obtained. Dividing the union of lower approximations by the total number of objects in the dataset produces a value in the range [0,1] that indicates how well this feature subset represents the original full feature set. This measure is termed the dependency degree in rough set theory and it is used as the evaluation function within feature selectors to perform data reduction (Jensen & Shen, 2004a; Swiniarski & Skowron, 2003).

Discernibility Matrix-Based Reduction

Many applications of rough sets to feature selection make use of discernibility matrices for finding reducts; for example, pattern recognition (Swiniarski & Skowron, 2003). A discernibility matrix is generated by comparing each object *i* with every other object *j* in a dataset and recording in entry (*i*,*j*) those features that differ. For finding reducts, the decision-relative discernibility matrix is of more interest. This only considers those object discernibilities that occur when the corresponding decision features differ. From this, the discernibility function can be defined - a concise notation of how each object within the dataset may be distinguished from the others. By finding the set of all prime implicants (i.e. minimal coverings) of the discernibility function, all the reducts of a system may be determined (Skowron & Rauszer, 1992).

Extensions

Variable precision rough sets (VPRS) (Ziarko, 1993) extends rough set theory by the relaxation of the subset operator. It was proposed to analyze and identify data patterns which represent statistical trends rather than functional. The main idea of VPRS is to allow objects to be classified with an error smaller than a certain predefined level. This introduced threshold relaxes the rough set notion of requiring no information outside the dataset itself, but facilitates extra flexibility when considering noisy data.

The reliance on discrete data for the successful operation of RST can be seen as a significant drawback of the approach. Indeed, this requirement of RST implies an objectivity in the data that is simply not present. For example, in a medical dataset, values such as *Yes* or *No* cannot be considered objective for a *Headache* attribute as it may not be straightforward to decide whether a person has a headache or not to a high degree of accuracy. Again, consider an attribute *Blood Pressure*. In the real world, this is a real-valued measurement but for the purposes of RST must be discretized into a small set of labels such as *Normal*, *High*, etc. Subjective judgments are required for establishing boundaries for objective measurements.

In the rough set literature, there are two main ways of handling continuous attributes – through fuzzy-rough sets and tolerance rough sets. Both approaches replace the traditional equivalence classes of crisp rough set theory with alternatives that are better suited to dealing with this type of data.

In the fuzzy-rough case, fuzzy equivalence classes are employed within a fuzzy extension of rough set theory, resulting in a hybrid approach (Jensen & Shen, 2007). Subjective judgments are not entirely removed as fuzzy set membership functions still need to be defined. However, the method offers a high degree of flexibility when dealing with real-valued data, enabling the vagueness and imprecision present to be modeled effectively (Dubois & Prade, 1992; De Cock et al., 2007; Yao, 1998). Data reduction methods based on this have been investigated with some success (Jensen & Shen, 2004a; Shen & Jensen, 2004; Yeung et al., 2005).

In the tolerance case, a measure of feature value similarity is employed and the lower and upper approximations defined based on these similarity measures. Such lower and upper approximations define tolerance rough sets (Skowron & Stepaniuk, 1996). By relaxing the transitivity constraint of equivalence classes, a further degree of flexibility (with regard to indiscernibility) is introduced. In traditional rough sets, objects are grouped into equivalence classes if their attribute values are equal. This requirement might be too strict for real-world data, where values might differ only as a result of noise. Methods based on tolerance rough

sets provide more flexibility than traditional rough set approaches, but are not as useful as fuzzy-rough set methods due to the dependency on crisp granulations

FUTURE TRENDS

Rough set theory will continue to be applied to data reduction as it possesses many essential characteristics for the field. For example, it requires no additional information other than the data itself and provides constructions and techniques for the effective removal of redundant or irrelevant features. In particular, developments of its extensions will be applied to this area in order to provide better tools for dealing with high-dimensional noisy, continuous-valued data. Such data is becoming increasingly common in areas as diverse as bioinformatics, visualization, microbiology, and geology.

CONCLUSION

This chapter has provided an overview of rough set-based approaches to data reduction. Current methods tend to concentrate on alternative evaluation functions, employing rough set concepts to gauge subset suitability. These methods can be categorized into two distinct approaches: those that incorporate the degree of dependency measure (or extensions), and those that apply heuristic methods to generated discernibility matrices.

Methods based on traditional rough set theory do not have the ability to effectively manipulate continuous data. For these methods to operate, a discretization step must be carried out beforehand, which can often result in a loss of information. There are two main extensions to RST that handle this and avoid information loss: tolerance rough sets and fuzzy-rough sets. Both approaches replace crisp equivalence classes with alternatives that allow greater flexibility in handling object similarity.

REFERENCES

Dash, M., & Liu, H. (1997). Feature Selection for Classification. *Intelligent Data Analysis*, 1(3), pp. 131-156.

Dubois, D., & Prade, H. (1992). Putting rough sets and fuzzy sets together. *Intelligent Decision Support*. Kluwer Academic Publishers, Dordrecht, pp. 203–232.

De Cock M., Cornelis C., & Kerre E.E. (2007). Fuzzy Rough Sets: The Forgotten Step. *IEEE Transactions on Fuzzy Systems*, 15(1), pp.121-130.

Jensen, R., & Shen, Q. (2004a). Semantics-Preserving Dimensionality Reduction: Rough and Fuzzy-Rough Based Approaches. *IEEE Transactions on Knowledge and Data Engineering*, 16(12), pp.1457-1471.

Jensen, R., & Shen, Q. (2004b). Fuzzy-Rough Attribute Reduction with Application to Web Categorization. *Fuzzy Sets and Systems*, 141(3), pp. 469-485.

Jensen, R., & Shen, Q. (2007). Fuzzy-Rough Sets Assisted Attribute Selection. *IEEE Transactions on Fuzzy Systems*, 15(1), pp.73-89.

Langley, P. (1994). Selection of relevant features in machine learning. In *Proceedings of the AAAI Fall Symposium on Relevance*, pp. 1-5.

Pawlak, Z. (1991). *Rough Sets: Theoretical Aspects of Reasoning about Data*. Kluwer Academic Publishing, Dordrecht.

Pawlak, Z. (2003). Some Issues on Rough Sets. *LNCS Transactions on Rough Sets*, vol. 1, pp. 1-53.

Polkowski, L. (2002). Rough Sets: Mathematical Foundations. *Advances in Soft Computing*. Physica Verlag, Heidelberg, Germany.

Shen, Q. & Jensen, R. (2004). Selecting Informative Features with Fuzzy-Rough Sets and Its Application for Complex Systems Monitoring. *Pattern Recognition* 37(7), pp. 1351–1363.

Siedlecki, W., & Sklansky, J. (1988). On automatic feature selection. *International Journal of Pattern Recognition and Artificial Intelligence, 2*(2), 197-220.

Skowron, A., & Rauszer, C. (1992). The discernibility matrices and functions in information systems. *Intelligent Decision Support*, Kluwer Academic Publishers, Dordrecht, pp. 331–362.

Skowron, A , & Grzymala-Busse, J. W. (1994) From rough set theory to evidence theory. In *Advances in the Dempster-Shafer Theory of Evidence*, (R. Yager, M. Fedrizzi, and J. Kasprzyk eds.), John Wiley & Sons Inc., 1994.

Skowron, A., & Stepaniuk, J. (1996). Tolerance Approximation Spaces. *Fundamenta Informaticae*, 27(2), 245–253.

Skowron, A., Pawlak, Z., Komorowski, J., & Polkowski, L. (2002). A rough set perspective on data and knowledge. *Handbook of data mining and knowledge discovery*, pp. 134-149, Oxford University Press.

Swiniarski, R.W., & Skowron, A. (2003). Rough set methods in feature selection and recognition. *Pattern Recognition Letters*, 24(6), 833-849.

Wygralak, M. (1989) Rough sets and fuzzy sets – some remarks on interrelations. *Fuzzy Sets and Systems*, 29(2), 241-243.

Yao, Y. Y. (1998). A Comparative Study of Fuzzy Sets and Rough Sets. *Information Sciences*, 109(1-4), 21–47.

Yeung, D. S., Chen, D., Tsang, E. C. C., Lee, J. W. T., & Xizhao, W. (2005). On the Generalization of Fuzzy Rough Sets. *IEEE Transactions on Fuzzy Systems*, 13(3), pp. 343–361.

Ziarko, W. (1993). Variable Precision Rough Set Model. *Journal of Computer and System Sciences*, 46(1), 39–59.

KEY TERMS

Core: The intersection of all reducts (i.e. those features that appear in all reducts). The core contains those features that cannot be removed from the data without introducing inconsistencies.

Data Reduction: The process of reducing data dimensionality. This may result in the loss of the semantics of the features through transformation of the underlying values, or, as in feature selection, may preserve their meaning. This step is usually carried out in order to visualize data trends, make data more manageable, or to simplify the resulting extracted knowledge.

Dependency Degree: The extent to which the decision feature depends on a given subset of features, measured by the number of discernible objects divided by the total number of objects.

Discernibility Matrix: Matrix indexed by pairs of object number, whose entries are sets of features that differ between objects.

Feature Selection: The task of automatically determining a minimal feature subset from a problem domain while retaining a suitably high accuracy in representing the original features, and preserving their meaning.

Fuzzy-Rough Sets: An extension of RST that employs fuzzy set extensions of rough set concepts to determine object similarities. Data reduction is achieved through use of fuzzy lower and upper approximations.

Lower Approximation: The set of objects that definitely belong to a concept, for a given subset of features.

Reduct: A subset of features that results in the maximum dependency degree for a dataset, such that no feature can be removed without producing a decrease in this value. A dataset may be reduced to those features occurring in a reduct with no loss of information according to RST.

Rough Set: An approximation of a vague concept, through the use of two sets – the lower and upper approximations.

Tolerance Rough Sets: An extension of RST that employs object similarity as opposed to object equality (for a given subset of features) to determine lower and upper approximations.

Upper Approximation: The set of objects that possibly belong to a concept, for a given subset of features.

Variable Precision Rough Sets: An extension of RST that relaxes the notion of rough set lower and upper approximations by introducing a threshold of permissible error.

Data Streams

João Gama
University of Porto, Portugal

Pedro Pereira Rodrigues
University of Porto, Portugal

INTRODUCTION

Nowadays, data bases are required to store massive amounts of data that are continuously inserted, and queried. Organizations use decision support systems to identify potential useful patterns in data. Data analysis is complex, interactive, and exploratory over very large volumes of historic data, eventually stored in distributed environments.

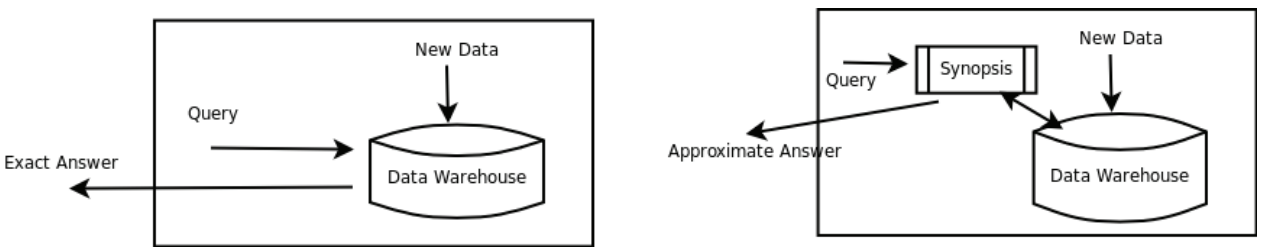
What distinguishes current data sources from earlier ones are the continuous flow of data and the automatic data feeds. We do not just have people who are entering information into a computer. Instead, we have computers entering data into each other (Muthukrishnan, 2005). Some illustrative examples of the magnitude of today data include: 3 billion telephone calls per day, 4 Giga Bytes of data from radio telescopes every night, 30 billion emails per day, 1 billion SMS, 5 Giga Bytes of Satellite Data per day, 70 billion IP Network Traffic per day. In these applications, data is modelled best not as persistent tables but rather as transient data streams. In some applications it is not feasible to load the arriving data into a traditional Data Base Management Systems (DBMS), and traditional DBMS are not designed to directly support the continuous queries required in these applications (Alon et al., 1996; Babcock et al. 2002; Cormode & Muthukrishnan, 2003). These sources of data are called *Data Streams*.

Computers play a much more active role in the current trends in decision support and data analysis. Data mining algorithms search for hypothesis, evaluate and suggest patterns. The pattern discovery process requires online ad-hoc queries, not previously defined, that are successively refined. Due to the exploratory nature of these queries, an exact answer may be not required: a user may prefer a fast but approximate answer to a exact but slow answer. Processing queries in streams require radically different type of algorithms. Range queries and selectivity estimation (the proportion of tuples that satisfy a query) are two illustrative examples where fast but approximate answers are more useful than slow and exact ones. Approximate answers are obtained from small data structures (synopsis) attached to data base that summarize information and can be updated incrementally. The general schema is presented in Figure 1.

An Illustrative Problem

A problem that clear illustrates the issues in streaming process (Datar, 2002), is the problem of finding the

Figure 1. Querying schemas: slow and exact answer vs fast but approximate answer using synopsis



maximum value (MAX) or the minimum value (MIN) in a sliding window over a sequence of numbers. When we can store in memory all the elements of the sliding window, the problem is trivial and we can find the exact solution. When the size of the sliding window is greater than the available memory, there is no exact solution. For example, suppose that the sequence is monotonically decreasing and the aggregation function is MAX. Whatever the window size, the first element in the window is always the maximum. As the sliding window moves, the exact answer requires maintaining all the elements in memory.

BACKGROUND

What makes Data Streams different from the conventional relational model? A key idea is that operating in the data stream model does not preclude the use of data in conventional stored relations. Some relevant differences (Babcock et al., 2002) include:

- The data elements in the stream arrive online.
- The system has no control over the order in which data elements arrive, either within a data stream or across data streams.
- Data streams are potentially unbound in size.
- Once an element from a data stream has been processed it is discarded or archived. It cannot be retrieved easily unless it is explicitly stored in memory, which is small relative to the size of the data streams.

In the streaming model (Muthukrishnan, 2005) the input elements $\mathbf{a_1}$, $\mathbf{a_2}$, **...** arrive sequentially, item by item, and describe an underlying function **A**. Streaming models differ on how $\mathbf{a_i}$ describe **A**. Three different models are:

- *Time Series Model*: once an element $\mathbf{a_i}$ is seen, it can not be changed.
- *Turnstile Model:* each $\mathbf{a_i}$ is an update to **A(j)**.
- *Cash Register Model:* each $\mathbf{a_i}$ is an increment **to** $\mathbf{A(j) = A(j)+a_i}$.

Research Issues in Data Streams Management Systems

Data streams are unbounded in length. However, this is not the only problem. The domain of the possible values of an attribute is also very large. A typical example is the domain of all pairs of IP addresses on the Internet. It is so huge, that makes exact storage intractable, as it is impractical to store all data to execute queries that reference past data. *Iceberg queries* process relative large amount of data to find aggregated values above some specific threshold, producing results that are often very small in number (the tip of the iceberg). Some query operators are unable to produce the first tuple of the output before they have seen the entire input. They are referred as *blocking query operators* (Babcock et al., 2002). Examples of blocking query operators are aggregating operators, like SORT, SUM, COUNT, MAX, MIN, join between multiple streams, etc. In the stream setting, continuous queries using block operators are problematic. Their semantics in the stream context is an active research area.

There are two basic approaches: sliding windows (Datar & Motwani, 2007) and summaries (Aggarwal & Yu, 2007. In the sliding windows approach a *time stamp* is associated with each tuple. The time stamp defines when a specific tuple is valid (e.g. inside the sliding window) or not. Queries run over the tuples inside the window. In the case of join multiple streams the semantics of timestamps is much less clear. For example, what is the time stamp of an output tuple?

In the latter, queries are executed over a summary, a compact data-structure that captures the distribution of the data. This approach requires techniques for storing summaries or synopsis information about previously seen data. Large summaries provide more precise answers. There is a trade-off between the size of summaries and the overhead to update summaries and the ability to provide precise answers. Histograms and wavelets are the most common used summaries, and are discussed later in this chapter.

From the point of view of a Data Stream Management System several research issues emerge. Illustrative problems (Babcock et al., 2002; Datar et al., 2002) include:

- Approximate query processing techniques to evaluate queries that require unbounded amount of memory.

Table 1. The following table summarizes the main differences between traditional and stream data processing

	Traditional	Stream
Number of passes	Multiple	Single
Processing Time	Unlimited	Restricted
Used Memory	Unlimited	Restricted
Result	Accurate	Approximate

- Sliding window query processing both as an approximation technique and as an option in the query language.
- Sampling to handle situations where the flow rate of the input stream is faster than the query processor.
- The meaning and implementation of blocking operators (e.g. aggregation and sorting) in the presence of unending streams.

MAIN FOCUS

The basic general bounds on the tail probability of a random variable include the Markov, Chebyshev and Chernoff inequalities (Motwani & Raghavan, 1997) that give the probability that a random variable deviates greatly from its expectation. The challenge consists of using sub-linear space, obtaining approximate answers with error guaranties. One of the most used approximate schemas is the so-called (ε, δ): Given two small positive numbers, ε, δ, compute an estimate that with probability 1- δ, is within a relative error less than ε.

A useful mathematical tool used elsewhere in solving the above problems is the so called **frequency moments**. A frequency moment is a number F^K, defined as

$$F^K = \sum_{i=1}^{v} m_i^K$$

where **v** is the domain size, $\mathbf{m}_i$ is the frequency of **i** in the sequence, and k>= 0. $\mathbf{F^0}$ is the number of distinct values in the sequence, $\mathbf{F^1}$ is the length of the sequence, $\mathbf{F^2}$ is known as the self-join size (the repeat rate or Gini's index of homogeneity). The frequency moments provide useful information about the data and can be used in query optimizing. In (Alon et al., 1996) the authors present an unexpected result that the second frequency moment of a stream of values can be approximated using logarithmic space. The authors also show that no analogous result holds for higher frequency moments.

Hash functions are another powerful tool in streaming process. They are used to project attributes with huge domains into lower space dimensions. One of the earlier results is the **Hash sketches** for **distinct-value counting** (aka **FM**) introduced by Flajolet & Martin (1983). The basic assumption is the existence of a hash function h(x) that maps incoming values x in (0,…, N-1) *uniformly* across (0,…, 2^L-1), where L = O(logN). The FM sketches are used for estimating the number of distinct items in a database (or stream) in one pass while using only a small amount of space.

Data Synopsis

With new data constantly arriving even as old data is being processed, the amount of computation time per data element must be low. Furthermore, since we are limited to bound amount of memory, it may not be possible to produce exact answers. High-quality approximate answers can be an acceptable solution. Two types of techniques can be used: data reduction and sliding windows. In both cases, they must use data structures that can be maintained incrementally. The most common used techniques for data reduction involve: sampling, synopsis and histograms, and wavelets.

Sampling

Instead of dealing with an entire data stream, we can sample instances at periodic intervals. The **reservoir sampling** technique (Vitter, 1985) is the classic algorithm to maintain an online random sample. The basic idea consists of maintaining a sample of size **s**, called the *reservoir*. As the stream flows, every new element

has a certain probability of replacing an old element in the reservoir. Extensions to maintain a sample of size *k* over a count-based sliding window of the *n* most recent data items from data streams appear in (Babcock et al., 2002a).

Synopsis and Histograms

Synopsis and Histograms are summarization techniques that can be used to approximate the frequency distribution of element values in a data stream. They are commonly used to capture attribute value distribution statistics for query optimizers (like range queries). A histogram is defined by a set of non-overlapping intervals. Each interval is defined by the boundaries and a frequency count. The reference technique is the V-Optimal histogram (Guha et al., 2004). It defines intervals that minimize the frequency variance within each interval. Sketches are special case of synopsis, which provide probabilistic guarantees on the quality of the approximate answer (e.g. the answer is 10±2 with probability 95%). Sketches have been used to solve the **k-hot items** (Cormode & Muthukrishnan 2003).

Wavelets

Wavelet transforms are mathematical techniques, in which signals are represented as a weighted sum of waveforms that attempt to capture trends in numerical functions. Wavelet analysis (Gilbert et al. 2003) is popular in several streaming applications, because most signals can be represented using a small set of coefficients. The simplest and most common transformation is the Haar wavelet (Jawerth & Sweldens, 1994). Any sequence (x_0, x_1,..., x_{2n}, x_{2n+1}) of even length is transformed into a sequence of two-component-vectors ((s_0, d_0 ,..., (s_n, d_n)). One stage of the Fast Haar-Wavelet Transform consists of:

$$\begin{bmatrix} s_i \\ d_i \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \mathrm{x} \begin{bmatrix} \mathrm{x_i} \\ x_{i+1} \end{bmatrix}$$

The process continues, by separating the sequences **s** and **d** and transforming the sequence **s**.

FUTURE TRENDS

Emerging topics in data stream research involve the semantics of block-operators in the stream scenario; join over multi-streams, and processing distributed streams with privacy preserving concerns. The inductive machine learning community study computer programs able to extract rules and patterns from massive data sets (Hulten & Domingos, 2001). Some decision models (decision trees, decision rules, etc.) can provide concise, compact, and interpretable description of massive data sets. Moreover, the summaries and synopsis generated from massive flows of data used in DSMS are relevant for the Data Mining community. Future research points to a symbiosis between research in DSMS and Machine Learning.

CONCLUSION

Data Stream Management Systems development is an emergent topic in the computer science community, as they present solutions to process and query massive and continuous flow of data. Algorithms that process data streams deliver approximate solutions, providing a fast answer using few memory resources. In some applications, mostly database oriented, an approximate answer should be within an admissible error margin. In general, as the range of the error decreases the space of computational resources goes up. The challenge is to find sub-linear space data-structures that provide approximate answers to queries with narrow error bounds.

REFERENCES

Aggarwal,, C. C., & Yu, P. S. (2007). A Survey of Synopsis Construction in Data Streams. In C. Aggarwal (Ed.), *Data Streams: Models and Algorithms* (pp. 169-207). Springer

Alon, N., Matias, Y., & Szegedy, M. (1996). The space complexity of approximating the frequency moments. In *Proceedings of the 28th ACM Symposium on Theory of Computing* (pp. 20-29). ACM Press.

Babcock, B., Datar, M., & Motwani, R. (2002). Sampling from a moving window over streaming data. In *Proceedings of the 13th Annual ACM-SIAM Symposium*

on Discrete Algorithms (pp. 633-634). Society for Industrial and Applied Mathematics.

Babcock, B., Babu, S., Datar, M., Motwani., R., & Widom, J. (2002). Models and issues in data stream systems. In *Proceedings of the 21th Symposium on Principles of Database Systems* (pp.1-16). ACM Press.

Cormode, G., & Muthukrishnan, S. (2003). What's hot and what's not: tracking most frequent items dynamically. In *Proceedings of the 22nd Symposium on Principles of Database Systems* (pp. 296-306). ACM Press.

Datar, M., Gionis, A., Indyk, P., & Motwani, R. (2002). Maintaining stream statistics over sliding windows. In *Proceedings of 13th Annual ACM-SIAM Symposium on Discrete Algorithms* (pp. 635-644). Society for Industrial and Applied Mathematics.

Datar, M., & Motwani, R. (2007). The Sliding Window Computation Model and Results. In C. Aggarwal (Ed.), *Data Streams: Models and Algorithms* (pp. 149-167). Springer

Flajolet, P., & Martin, G. N. (1983). Probabilistic Counting. In *Proceedings of the 24th Symposium on Foundations of Computer Science* (pp.76-82). IEEE Computer Society.

Gilbert, A. C., Kotidis, Y., Muthukrishnan, S., & Strauss, M. J. (2003). One-pass wavelet decompositions of data streams. *IEEE Transactions on Knowledge and Data Engineering, 15* (3), 541-554.

Guha, S., Rastogi, R., & Shim, K. (1998). CURE: An efficient clustering algorithm for large databases. In L. Haas and A. Tiwary (Eds.), *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data* (pp. 73-84). ACM Press.

Guha, S., Shim, K., & Woo, J. (2004). REHIST: Relative error histogram construction algorithms. In *Proceedings of International Conference on Very Large Data Bases* (pp. 300-311). Morgan Kaufmann.

Hulten, G., & Domingos, P. (2001). Catching up with the data: research issues in mining data streams. In *Proceedings of Workshop on Research issues in Data Mining and Knowledge Discovery*.

Jawerth, B., & Sweldens, W. (1994). An overview of wavelet based multiresolution analysis. *SIAM Review, 36* (3), 377-412.

Motwani, R., & Raghavan, P. (1997). *Randomized Algorithms*. Cambridge University Press.

Muthukrishnan, S. (2005). *Data streams: algorithms and applications*. Now Publishers.

Vitter, J. S. (1985). Random sampling with a reservoir. *ACM Transactions on Mathematical Software, 11* (1), 37-57.

KEY TERMS

Blocking Query Operators: Query operators that only produce the first output tuple after seeing the entire input.

Data Mining: Process of extraction of useful information in large Data Bases.

Data Stream: Continuous flow of data eventually at high speed

Histograms: Graphical display of tabulated frequencies.

Iceberg Queries: Queries that compute aggregate functions over an attribute to find aggregate values over some specific threshold.

Machine Learning: Programming computers to optimize a performance criterion using example data or past experience.

Wavelet: Representation of a signal in terms of a finite length or fast decaying oscillating waveform (known as the *mother wavelet*).

Sampling: Subset of observations about a population.

Summaries: Compact data-structure that capture the distribution of data. (see Histograms, Wavelets, Synopsis).

Synopsis: Summarization technique that can be used to approximate the frequency distribution of element values in a data stream.

Data Transformation for Normalization

Amitava Mitra
Auburn University, USA

INTRODUCTION

As the abundance of collected data on products, processes and service-related operations continues to grow with technology that facilitates the ease of data collection, it becomes important to use the data adequately for decision making. The ultimate value of the data is realized once it can be used to derive information on product and process parameters and make appropriate inferences.

Inferential statistics, where information contained in a sample is used to make inferences on unknown but appropriate population parameters, has existed for quite some time (Mendenhall, Reinmuth, & Beaver, 1993; Kutner, Nachtsheim, & Neter, 2004). Applications of inferential statistics to a wide variety of fields exist (Dupont, 2002; Mitra, 2006; Riffenburgh, 2006).

In data mining, a judicious choice has to be made to extract observations from large databases and derive meaningful conclusions. Often, decision making using statistical analyses requires the assumption of normality. This chapter focuses on methods to transform variables, which may not necessarily be normal, to conform to normality.

BACKGROUND

With the normality assumption being used in many statistical inferential applications, it is appropriate to define the normal distribution, situations under which non-normality may arise, and concepts of data stratification that may lead to a better understanding and inference-making. Consequently, statistical procedures to test for normality are stated.

Normal Distribution

A continuous random variable, Y, is said to have a normal distribution, if its probability density function is given by the equation

$$f(y) = \frac{1}{\sqrt{2\pi}\sigma} \exp[-(y-\mu)^2 / 2\sigma^2], \quad (1)$$

where μ and σ denote the mean and standard deviation, respectively, of the normal distribution. When plotted, equation (1) resembles a bell-shaped curve that is symmetric about the mean (μ). A cumulative distribution function (cdf), F(y), represents the probability $P[Y \leq y]$, and is found by integrating the density function given by equation (1) over the range $(-\infty, y)$. So, we have the cdf for a normal random variable as

$$F(y) = \int_{-\infty}^{y} \frac{1}{\sqrt{2\pi}\sigma} \exp[-(x-\mu)^2 / 2\sigma^2]dx. \quad (2)$$

In general, $P[a \leq Y \leq b] = F(b) - F(a)$.

A standard normal random variable, Z, is obtained through a transformation of the original normal random variable, Y, as follows:

$$Z = (Y - \mu)/\sigma . \quad (3)$$

The standard normal variable has a mean of 0 and a standard deviation of 1 with its cumulative distribution function given by F(z).

Non-Normality of Data

Prior to analysis of data, careful consideration of the manner in which the data is collected is necessary. The following are some considerations that data analysts should explore as they deal with the challenge of whether the data satisfies the normality assumption.

Data Entry Errors

Depending on the manner in which data is collected and recorded, data entry errors may highly distort the distribution. For instance, a misplaced decimal point on an observation may lead that observation to become an outlier, on the low or the high side. **Outliers** are observations that are "very large" or "very small"

compared to the majority of the data points and have a significant impact on the **skewness** of the distribution. Extremely large observations will create a distribution that is right-skewed, whereas outliers on the lower side will create a negatively-skewed distribution. Both of these distributions, obviously, will deviate from normality. If outliers can be justified to be data entry errors, they can be deleted prior to subsequent analysis, which may lead the distribution of the remaining observations to conform to normality.

Grouping of Multiple Populations

Often times, the distribution of the data does not resemble any of the commonly used statistical distributions let alone normality. This may happen based on the nature of what data is collected and how it is grouped. Aggregating data that come from different populations into one dataset to analyze, and thus creating one "superficial" population, may not be conducive to statistical analysis where normality is an associated assumption. Consider, for example, the completion time of a certain task by operators who are chosen from three shifts in a plant. Suppose there are inherent differences between operators of the three shifts, whereas within a shift the performance of the operators is homogeneous. Looking at the aggregate data and testing for normality may not be the right approach. Here, we may use the concept of **data stratification** and subdivide the aggregate data into three groups or populations corresponding to each shift.

Parametric versus Nonparametric Tests

While data from many populations may not necessarily be normal, one approach for dealing with this problem, when conducting parametric tests, is to determine a suitable transformation such that the transformed variable satisfies the normality assumption. Alternatively, one may consider using nonparametric statistical tests (Conover, 1999; Daniel, 1990) for making inferences. The major advantage of nonparametric tests is that they do not make any assumption on the form of the distribution. Hence, such tests could be used for data that are not from normal distributions. There are some disadvantages to nonparametric tests however. One significant disadvantage deals with the **power** of the test. The power of a statistical test is its ability to identify and reject a null hypothesis when the null is false. If the assumptions associated with a parametric test are satisfied, the power of the parametric test is usually larger than that of its equivalent nonparametric test. This is the main reason for the preference of a parametric test over an equivalent nonparametric test.

Validation of Normality Assumption

Statistical procedures known as **goodness-of-fit tests** make use of the empirical cumulative distribution function (cdf) obtained from the sample versus the theoretical cumulative distribution function, based on the hypothesized distribution. Moreover, parameters of the hypothesized distribution may be specified or estimated from the data. The test statistic could be a function of the difference between the observed frequency, and the expected frequency, as determined on the basis of the distribution that is hypothesized. Goodness-of-fit tests may include chi-squared tests (Duncan, 1986), Kolmogorov-Smirnov tests (Massey, 1951), or the Anderson-Darling test (Stephens, 1974), among others. Along with such tests, graphical methods such as **probability plotting** may also be used.

Probability Plotting

In probability plotting, the sample observations are ranked in ascending order from smallest to largest. Thus, the observations $x_1, x_2, \ldots, x_n$ are ordered as $x_{(1)}, x_{(2)}, \ldots, x_{(n)}$, where $x_{(1)}$ denotes the smallest observation and so forth. The **empirical cumulative distribution function (cdf)** of the ith ranked observation, $x_{(i)}$, is given by

$$F_i = \frac{i - 0.5}{n}. \tag{4}$$

The **theoretical cdf**, based on the hypothesized distribution, at $x_{(i)}$, is given by $G(x_{(i)})$, where $G(\cdot)$ is calculated using specified parameters or estimates from the sample. A probability plot displays the plot of $x_{(i)}$, on the horizontal axis, versus F_i and $G(x_{(i)})$ on the vertical axis. The **vertical axis** is **so scaled** such that if the data is from the hypothesized distribution, say normal, the plot of $x_{(i)}$ versus $G(x_{(i)})$ will be a straight line. Thus, departures of $F(\cdot)$ from $G(\cdot)$ are visually easy to detect. The closer the plotted values of $F(\cdot)$ are to the fitted line, $G(\cdot)$, the stronger the support for the null hypothesis. A test statistic is calculated where large

deviations between F(·) and G(·) lead to large values of the test statistic, or alternatively small **p-values**. Hence, if the observed **p-value** is less than α, the chosen level of significance, the null hypothesis representing the hypothesized distribution is rejected.

In a **normal probability plot**, suppose we are testing the null hypothesis that the data is from a normal distribution, with mean and standard deviation not specified. Then, $G(x_{(i)})$ will be $\Phi((x_{(i)} - \bar{x})/s)$, $\Phi(\cdot)$ represents the cdf of a standard normal distribution and $\bar{x}$ and s are sample estimates of the mean and standard deviation respectively.

The software, Minitab (Minitab, 2007), allows probability plotting based on a variety of chosen distributions such as normal, lognormal, exponential, gamma, or Weibull. A common test statistic used for such tests is the Anderson-Darling statistic which measures the area between the fitted line (based on the hypothesized distribution) and the empirical cdf. This statistic is a squared distance that is weighted more heavily in the tails of the distribution. Smaller values of this statistic lead to the non-rejection of the null hypothesis and confirm validity of the observations being likely from the hypothesized distribution. Alternative tests for normality also exist (Shapiro & Wilk, 1965).

MAIN FOCUS

In this section we describe possible ways to transform the original variable such that the transformed variable may satisfy the assumption of normality. There is, of course, no assurance that any variable can always be transformed to a normal random variable.

Discrete Variables

Let us first consider some discrete random variables and their associated transformations to normality. Such variables could represent attribute information, for example, the number of acceptable parts in a batch. Alternatively, it could represent a count of the number of events, for instance, the number of bacterial count per unit volume.

Binomial Random Variable

A Binomial random variable is one where each outcome could be one of two types, i.e., a part acceptable or not. The random variable represents the total number of outcomes of a certain type, say acceptable parts. Thus, in an automatic inspection station, where parts are produced in batches of size 500, the Binomial random variable could be the number of acceptable parts (Y) in each batch. Alternatively, this could be re-expressed as the proportion of acceptable parts (p) in each batch, whose estimate is given by $\hat{p} = y/n$, where n represents the batch size.

It is known that when p is not close to 0.5, or in other words when p is very small (near 0) or very large (close to 1), the distribution of the estimated proportion of acceptable parts is quite asymmetric and highly skewed and will not resemble normality. A possible transformation for Binomial data is the following:

$$Y_T = \text{arc sine}\left(\sqrt{\hat{p}}\right) \tag{5}$$

or

$$Y_T = \ell n\left[\frac{\hat{p}}{1-\hat{p}}\right] \tag{6}$$

The distribution of the transformed variable, Y_T, may approach normality.

Poisson Random Variable

A Poisson random variable represents the number of events (Y) that happen within a product unit, space or volume, or time period, where it is assumed that the events happen randomly and independently. Examples are the number of customer complaints on a monthly basis in a department store, or the number of people entering the intensive care unit in a hospital per week. A possible transformation in this context is the square root transformation given by

$$Y_T = \sqrt{Y} \tag{7}$$

Continuous Variables

Data on most physical variables are measured on a continuum scale. Conceptually, continuous variables can take on an infinite number of values within a given range. Examples are dimensional measurements, temperature, pressure, density, viscosity, and strength.

In transforming continuous variables, the shape of the distribution, as determined by its skewness, may provide an indication of the type of transformation. Let us first consider right-skewed distributions and identify modifications in this context.

Square Root Transformation

Such a transformation applies to data values that are positive. So, if there are negative values, a constant must be added to move the minimum value of the distribution above zero, preferably to 1. The rational for this being that for numbers of 1 and above, the square root transformation behaves differently than for numbers between 0 and 1. This transformation has the effect of reducing the length of the right tail and so is used to bring a right-tailed distribution closer to normality. The form of the transformation is given by

$$Y_T = \sqrt{Y + c} \qquad (8)$$

where c is a constant and such that min (Y + c) ≥ 1.

Log Transformation

Logarithmic transformations are also used for right-skewed distributions. Their impact is somewhat stronger than the square root transformation in terms of reducing the length of the right tail. They represent a class of transformations where the base of the logarithm can vary. Since the logarithm of a negative number is undefined, a translation must be made to the original variable if it has negative values. Similar to the square root transformation, it is desirable to shift the original variable such that the minimum is at 1 or above.

Some general guidelines for choosing the base of the logarithmic transformation exist. Usually, the base of 10, 2, and e should be considered as a minimum, where e = 2.7183 represents the base of natural logarithm. In considering the dataset, for a large range of the values, a base of 10 is desirable. Higher bases tend to pull extreme values in more drastically relative to lower bases. Hence, when the range of values is small, a base of e or 2 could be desirable. The logarithmic transformation is given by

$$Y_T = \log_a(Y + b) \ , \qquad (9)$$

where *a* represents a choice of the base of the logarithm and *b* is a constant such that

min (Y + b) ≥ 1.

Inverse Transformation

The inverse transformation is also used for a right-skewed distribution. Relative to the square root and logarithmic transformation, it has the strongest impact in terms of reducing the length of the right tail.

Taking the inverse makes very large numbers very small and very small numbers very large. It thus has the effect of reversing the order of the data values. So, in order to maintain the same ordering as the original values, the transformed values are multiplied by -1. The inverse transformed variable is given by

$$Y_T = -\frac{1}{Y} \qquad (10)$$

Since equation (10) is not defined for y = 0, a translation in the data could be made prior to the transformation in such a situation.

Power Transformations

We now consider the more general form of power transformations (Box & Cox, 1964; Sakia, 1992) where the original variable (Y) is raised to a certain power (p). The general power transformed variable (Y_T) is given by

$$\begin{aligned} Y_T &= Y^P, \ p > 0 \\ &= -(Y^P), \ p < 0 \\ &= \ell n\,(Y), \ p = 0 \end{aligned} \qquad (11)$$

The impact of the power coefficient, p, is on changing the distributional shape of the original variable. Values of the exponent, p > 1, shift weight to the upper tail of the distribution and reduce negative skewness. The larger the power, the stronger the effect. A parametric family of power transformations is known as a Box-Cox transformation (Box & Cox, 1964).

The three previously considered square root, logarithm, and inverse transformations are special cases of the general power transformation given by equation (11). Selection of the power coefficient (p) will be influenced by the type of data and its degree of skewness. Some guidelines are provided in a separate chapter.

Exponential Transformation

While the transformations given by equation (11) are used for positive values of the observations, another alternative (Manly, 1976) could be used in the presence of negative observations. This transformation is effective for a skewed unimodal distribution to be shifted to a symmetric normal-like distribution and is given by:

$$Y_T = [\exp(pY) - 1] / p, \text{ for } p \neq 0$$
$$= Y \quad , p = 0 \qquad (12)$$

FUTURE TRENDS

Testing hypothesis on population parameters through hypothesis testing or developing confidence intervals on these parameters will continue a trend of expanded usage. Applications of inferential statistics will continue to grow in virtually all areas in which decision making is based on data analysis. Thus, the importance of data transformations to meet the assumption of normality cannot be overemphasized.

CONCLUSION

The need for transformations when conducting data analyses is influenced by the assumptions pertaining to type of statistical test/analysis. Common assumptions may include independence and identical population distributions from which observations are chosen, common variance of two or more populations, and normality of population distribution. This paper has focused on satisfying the assumption of normality of the population distribution.

Prior to conducting data transformations, one should skim the data for possible recording or input errors or for identifiable special causes caused by circumstances that are not normally part of the process. If these events are identified, the data values can be deleted prior to analysis. However, it is of importance to not automatically delete extreme observations (large or small) unless a justifiable rationale has first been established.

REFERENCES

Box, G.E.P., & Cox, D.R. (1964). An analysis of transformations. *Journal of the Royal Statistical Society*, Series B, 26(2), 211-252.

Conover, W.J. (1999). *Practical Nonparametric Statistics*, 3rd edition, New York, Wiley.

Daniel, W.W. (1990). *Applied Nonparametric Statistics*, 2nd edition, Boston, MA, PWS-Kent.

Duncan, A.J. (1986). *Quality Control and Industrial Statistics*, 5th edition, Homewood, IL, Irwin.

Dupont, W.D. (2002). *Statistical Modeling for Biomedical Researchers*, New York, Cambridge University Press.

Kutner, M.H., Nachtsheim, C.J., & Neter, J. (2004). *Applied Linear Statistical Models*, Fifth edition, Homewood, IL, Irwin/McGraw-Hill.

Manly, B.F.J. (1976). Exponential data transformation. *The Statistician*, 25, 37-42.

Massey, F.J. (1951). The Kolmogorov-Smirnov test for goodness-of-fit. *Journal of the American Statistical Association*, 46, 68-78.

Mendenhall, W., Reinmuth, J.E., & Beaver, R.J. (1993). *Statistics for Management and Economics*, 7th edition, Belmont, CA, Duxbury.

Minitab. (2007). *Release 15*, State College, PA, Minitab.

Mitra, A. (2006). *Fundamentals of Quality Control and Improvement*, 2nd edition update, Mason, OH, Thomson.

Riffenburgh, R.H. (2006). *Statistics in Medicine*, 2nd edition, Burlington, MA, Elsevier Academic Press.

Sakia, R.M. (1992). The Box-Cox transformation techniques: A review. *The Statistician*, 41(2), 169-178.

Shapiro, S. & Wilk, M. (1965). An analysis of variance test for normality. *Biometrika*, 52, 591-611.

Stephens, M.A. (1974). EDI statistics for goodness-of-fit and some comparisons. *Journal of the American Statistical Association*, 69, 730-737.

KEY TERMS

Data Stratification: A scheme to partition aggregate data into groups or clusters based on the value of a selected variable(s).

Goodness-of-Fit Tests: A class of statistical tests that includes testing the distributional assumption based on comparing the observed frequency of data to the expected frequency under the hypothesized distribution.

Inferential Statistics: A field of statistics where decisions on unknown population parameters are made from information contained in samples.

Normal Distribution: A distribution, also known as the Gaussian distribution, whose density function is an exponential distribution symmetric about the mean. It has two parameters, the mean which is a measure of location and the standard deviation, a measure of dispersion. The equation of the density function is given by equation (1).

Outliers: Data values that are extremely large or extremely small compared to the majority of the values. Based on the distribution, for example normal distribution, an outlier may be considered as one that is more than three standard deviations away from the mean.

Parametric Tests: Statistical tests that deal with making inferences on parameters of distributions, for example estimation or hypothesis testing of the mean of a population. Such tests usually make assumptions on the distributional form.

Power of a Test: The power of a statistical test of hypothesis is the degree to which the test is able to identify and reject a null hypothesis, when the null hypothesis is false. A large power is desirable.

Probability Value: This refers to the chance of obtaining the data that has been observed, or something more extreme, if the null hypothesis is true. Also known as the p-value, it is used in hypothesis testing for decision making. If the p-value is less than the chosen level of significance, the null hypothesis is rejected.

Skewness of a Distribution: A measure of the degree of asymmetry of the distributional form, typically represented by a skewness coefficient. The degree of concentration of the data values along with the presence of outliers may influence skewness.

Data Warehouse Back-End Tools

Alkis Simitsis
National Technical University of Athens, Greece

Dimitri Theodoratos
New Jersey Institute of Technology, USA

INTRODUCTION

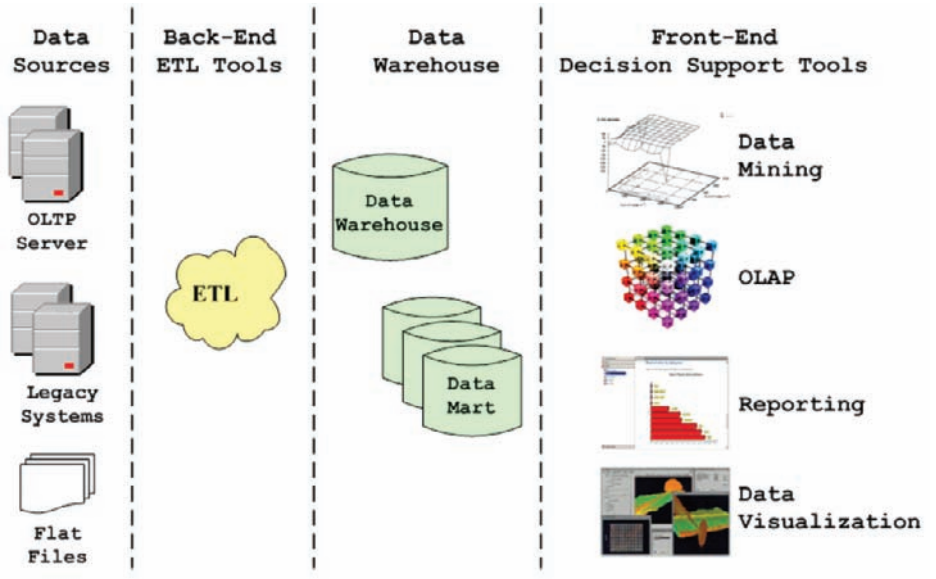
The back-end tools of a data warehouse are pieces of software responsible for the extraction of data from several sources, their cleansing, customization, and insertion into a data warehouse. In general, these tools are known as Extract – Transformation – Load (ETL) tools and the process that describes the population of a data warehouse from its sources is called ETL process. In all the phases of an ETL process (extraction and transportation, transformation and cleaning, and loading), individual issues arise, and, along with the problems and constraints that concern the overall ETL process, make its lifecycle a very complex task.

BACKGROUND

A Data Warehouse (DW) is a collection of technologies aimed at enabling the knowledge worker (executive, manager, analyst, etc.) to make better and faster decisions. The architecture of the data warehouse environment exhibits various layers of data in which data from one layer are derived from data of the previous layer (Figure 1).

The front-end layer concerns end-users who access the data warehouse with decision support tools in order to get insight into their data by using either advanced data mining and/or OLAP (On-Line Analytical Processing) techniques or advanced reports and visualizations. The central data warehouse layer comprises the data warehouse fact and dimension tables along with the

Figure 1. Abstract architecture of a Data Warehouse



appropriate application-specific data marts. The back stage layer includes all the operations needed for the collection, integration, cleaning and transformation of data coming from the sources. Finally, the sources layer consists of all the sources of the data warehouse; these sources can be in any possible format, such as OLTP (On-Line Transaction Processing) servers, legacy systems, flat files, xml files, web pages, and so on.

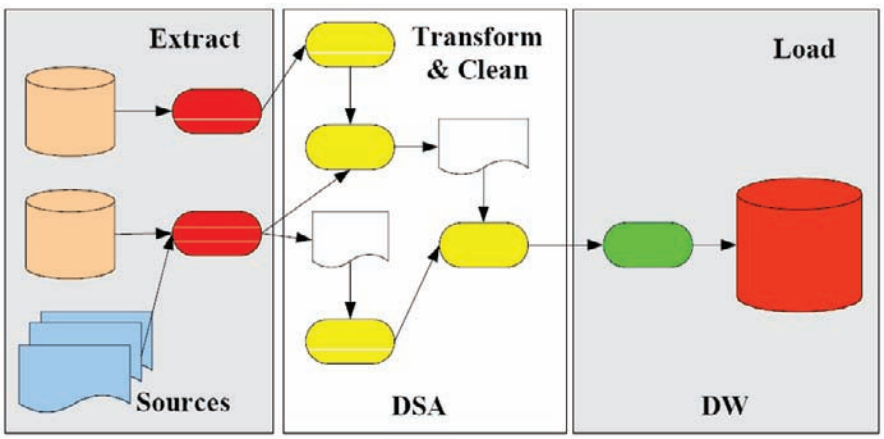
This article deals with the processes, namely ETL processes, which take place in the back stage of the data warehouse environment. The ETL processes are data intensive, complex, and costly (Vassiliadis, 2000). Several reports mention that most of these processes are constructed through an in-house development procedure that can consume up to 70% of the resources for a data warehouse project (Gartner, 2003). The functionality of these processes includes: (a) the identification of relevant information at the source side; (b) the extraction of this information; (c) the transportation of this information from the sources to an intermediate place called Data Staging Area (DSA); (d) the customization and integration of the information coming from multiple sources into a common format; (e) the cleaning of the resulting data set, on the basis of database and business rules; and (f) the propagation of the homogenized and cleansed data to the data warehouse and/or data marts. In the sequel, we will adopt the general acronym ETL for all kinds of in-house or commercial tools, and all the aforementioned categories of tasks.

Figure 2 abstractly describes the general framework for ETL processes. In the left side, the original data providers (Sources) exist. The data from these sources are extracted by extraction routines, which provide either complete snapshots or differentials of the data sources. Next, these data are propagated to the Data Staging Area (DSA) where they are transformed and cleaned before being loaded to the data warehouse. Intermediate results in the form of (mostly) files or relational tables are part of the data staging area. The data warehouse (DW) is depicted in the right part of Fig. 2 and comprises the target data stores, i.e., fact tables for the storage of information and dimension tables with the description and the multidimensional, roll-up hierarchies of the stored facts. The loading of the central warehouse is performed from the loading activities depicted right before the data warehouse data store.

State of the Art

In the past, there have been research efforts towards the design and optimization of ETL tasks. Among them, the following systems are dealing with ETL issues: (a) the AJAX system (Galhardas et al., 2000), (b) the Potter's Wheel system (Raman & Hellerstein, 2001), and (c) Arktos II (Arktos II, 2004). The first two prototypes are based on algebras, which are mostly tailored for the case of homogenizing web data; the latter concerns the modeling and the optimization of ETL processes in a customizable and extensible manner. Additionally, several research efforts have dealt with individual issues and problems of the ETL processes: (a) design and

Figure 2. The environment of Extract-Transform-Load processes



modeling issues (Luján-Mora et al., 2004; Skoutas and Simitsis, 2006; Trujillo J. and Luján-Mora, S., 2003; Vassiliadis et al., 2002), and (b) optimization issues (Simitsis et al., 2005; Tziovara et al., 2007). A first publicly available attempt towards the presentation of a benchmark for ETL Processes is proposed by Vassiliadis et al., 2007.

An extensive review of data quality problems and related literature, along with quality management methodologies can be found in Jarke et al. (2000). Rundensteiner (1999) offers a discussion of various aspects of data transformations. Sarawagi (2000) offers a similar collection of papers in the field of data including a survey (Rahm & Do, 2000) that provides an extensive overview of the field, along with research issues and a review of some commercial tools and solutions on specific problems, e.g., Monge (2000) and Borkar et al. (2000). In a related, but different, context, the IBIS tool (Calì et al., 2003) deals with integration issues following the global-as-view approach to answer queries in a mediated system.

In the market, there is a plethora of commercial ETL tools. Simitsis (2003) and Wikipedia (2007) contain a non-exhaustive list of such tools.

MAIN THRUST OF THE CHAPTER

In this section, we briefly review the problems and constraints that concern the overall ETL process, as well as the individual issues that arise in each phase of an ETL process (extraction and exportation, transformation and cleaning, and loading) separately. Simitsis (2004) offers a detailed study on the problems described in this chapter and presents a framework towards the modeling and the optimization of ETL processes.

Scalzo (2003) mentions that 90% of the problems in data warehouses rise from the nightly batch cycles that load the data. At this period, the administrators have to deal with problems like (a) efficient data loading, and (b) concurrent job mixture and dependencies. Moreover, ETL processes have global time constraints including their initiation time and their completion deadlines. In fact, in most cases, there is a tight 'time window' in the night that can be exploited for the refreshment of the data warehouse, since the source system is off-line or not heavily used during this period. Other general problems include: the scheduling of the overall process, the finding of the right execution order for dependent jobs and job sets on the existing hardware for the permitted time schedule, the maintenance of the information in the data warehouse.

Phase I: Extraction & Transportation

During the ETL process, a first task that must be performed is the extraction of the relevant information that has to be further propagated to the warehouse (Theodoratos et al., 2001). In order to minimize the overall processing time, this involves only a fraction of the source data that has changed since the previous execution of the ETL process, mainly concerning the newly inserted and possibly updated records. Usually, change detection is physically performed by the comparison of two snapshots (one corresponding to the previous extraction and the other to the current one). Efficient algorithms exist for this task, like the snapshot differential algorithms presented in (Labio & Garcia-Molina, 1996). Another technique is log 'sniffing', i.e., the scanning of the log file in order to 'reconstruct' the changes performed since the last scan. In rare cases, change detection can be facilitated by the use of triggers. However, this solution is technically impossible for many of the sources that are legacy systems or plain flat files. In numerous other cases, where relational systems are used at the source side, the usage of triggers is also prohibitive both due to the performance degradation that their usage incurs and the need to intervene in the structure of the database. Moreover, another crucial issue concerns the transportation of data after the extraction, where tasks like ftp, encryption-decryption, compression-decompression, etc., can possibly take place.

Phase II: Transformation & Cleaning

It is possible to determine typical tasks that take place during the transformation and cleaning phase of an ETL process. Rahm & Do (2000) further detail this phase in the following tasks: (a) data analysis; (b) definition of transformation workflow and mapping rules; (c) verification; (d) transformation; and (e) backflow of cleaned data. In terms of the transformation tasks, we distinguish two main classes of problems (Lenzerini, 2002): (a) conflicts and problems at the schema level (e.g., naming and structural conflicts), and, (b) data level transformations (i.e., at the instance level).

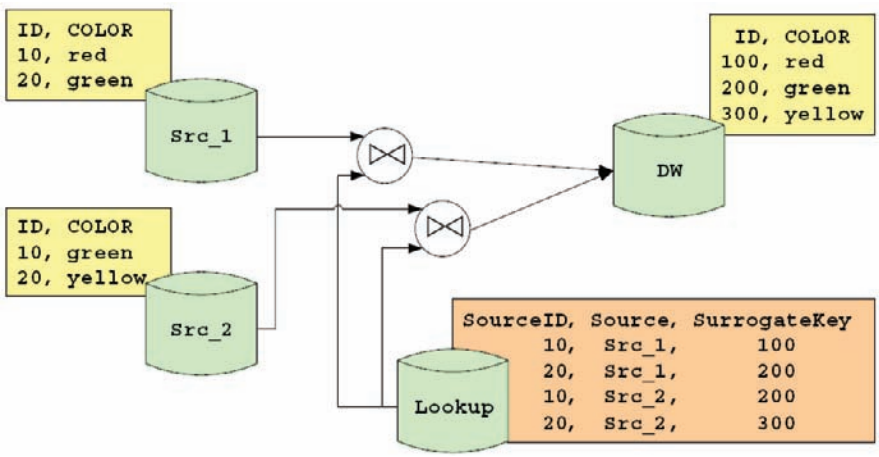
The integration and transformation programs perform a wide variety of functions, such as reformatting data, recalculating data, modifying key structures of data, adding an element of time to data warehouse data, identifying default values of data, supplying logic to choose between multiple sources of data, summarizing data, merging data from multiple sources, etc.

In the sequel, we present four common ETL transformation cases as examples: (a) semantic normalization and denormalization; (b) surrogate key assignment; (c) slowly changing dimensions; and (d) string problems. The research prototypes presented in the previous section and several commercial tools have already done some piece of progress in order to tackle with problems like these four. Still, their presentation in this article aspires to make the reader to understand that the whole process should be discriminated from the way we resolved integration issues until now.

Semantic normalization and denormalization. It is common for source data to be long denormalized records, possibly involving more than a hundred attributes. This is due to the fact that bad database design or classical COBOL tactics led to the gathering of all the necessary information for an application in a single table/file. Frequently, data warehouses are also highly denormalized, in order to answer more quickly certain queries. But, sometimes it is imperative to normalize somehow the input data. Consider, for example, a table of the form `R(KEY,TAX,DISCOUNT,PRICE)`, which we would like to transform to a table of the form `R'(KEY,CODE,AMOUNT)`. For example, the input tuple `t[key,30,60,70]` is transformed into the tuples `t1[key,1,30]; t2[key,2,60]; t3[key,3,70]`. The transformation of the information organized in rows to information organized in columns is called rotation or denormalization (since, frequently, the derived values, e.g., the total income in our case, are also stored as columns, functionally dependent on other attributes). Occasionally, it is possible to apply the reverse transformation, in order to normalize denormalized data before being loaded to the data warehouse.

Surrogate Keys. In a data warehouse project, we usually replace the keys of the production systems with a uniform key, which we call a *surrogate key* (Kimball et al., 1998). The basic reasons for this replacement are performance and semantic homogeneity. Performance is affected by the fact that textual attributes are not the best candidates for indexed keys and need to be replaced by integer keys. More importantly, semantic homogeneity causes reconciliation problems, since different production systems might use different keys for the same object (synonyms), or the same key for different objects (homonyms), resulting in the need for a global replacement of these values in the data warehouse. Observe row (`20,green`) in table `Src_1` of Figure 3. This row has a synonym conflict with row (`10,green`) in table `Src_2`, since they represent the same real-world entity with different `ID`'s, and a homonym conflict with row (`20,yellow`) in table `Src_2`

Figure 3. Surrogate key assignment



(over attribute ID). The *production key* ID is replaced by a *surrogate key* through a lookup table of the form Lookup(SourceID,Source,SurrogateKey). The Source column of this table is required because there can be synonyms in the different sources, which are mapped to different objects in the data warehouse (e.g., value 10 in tables Src_1 and Src_2). At the end of this process, the data warehouse table DW has globally unique, reconciled keys.

Slowly Changing Dimensions. Factual data are not the only data that change. The dimension values change at the sources, too, and there are several policies to apply for their propagation to the data warehouse. (Kimball et al., 1998) present three policies for dealing with this problem: overwrite (*Type* 1), create a new dimensional record (*Type* 2), push down the changed value into an "old" attribute (*Type* 3).

For *Type* 1 processing we only need to issue appropriate update commands in order to overwrite attribute values in existing dimension table records. This policy can be used whenever we do not want to track history in dimension changes and if we want to correct erroneous values. In this case, we use an old version of the dimension data D_{old} as they were received from the source and their current version D_{new}. We discriminate the new and updated rows through the respective operators. The new rows are assigned a new surrogate key, through a function application. The updated rows are assigned a surrogate key (which is the same as the one that their previous version had already being assigned). Then, we can join the updated rows with their old versions from the target table (which will subsequently be deleted) and project only the attributes with the new values.

In the *Type* 2 policy we copy the previous version of the dimension record and create a new one with a new surrogate key. If there is no previous version of the dimension record, we create a new one from scratch; otherwise, we keep them both. This policy can be used whenever we want to track the history of dimension changes.

Last, *Type* 3 processing is also very simple, since again we only have to issue update commands to existing dimension records. For each attribute A of the dimension table, which is checked for updates, we need to have an extra attribute called "old_A". Each time we spot a new value for A, we write the current A value to the old_A field and then write the new value to attribute A. This way we can have both new and old values present at the same dimension record.

String Problems. A major challenge in ETL processes is the cleaning and the homogenization of string data, e.g., data that stands for addresses, acronyms, names etc. Usually, the approaches for the solution of this problem include the application of regular expressions for the normalization of string data to a set of 'reference' values.

Phase III: Loading

The final loading of the data warehouse has its own technical challenges. A major problem is the ability to discriminate between new and existing data at loading time. This problem arises when a set of records has to be classified to (a) the new rows that need to be appended to the warehouse and (b) rows that already exist in the data warehouse, but their value has changed and must be updated (e.g., with an UPDATE command). Modern ETL tools already provide mechanisms towards this problem, mostly through language predicates. Also, simple SQL commands are not sufficient since the open-loop-fetch technique, where records are inserted one by one, is extremely slow for the vast volume of data to be loaded in the warehouse. An extra problem is the simultaneous usage of the rollback segments and log files during the loading process. The option to turn them off contains some risk in the case of a loading failure. So far, the best technique seems to be the usage of the batch loading tools offered by most RDBMS's that avoids these problems. Other techniques that facilitate the loading task involve the creation of tables at the same time with the creation of the respective indexes, the minimization of inter-process wait states, and the maximization of concurrent CPU usage.

FUTURE TRENDS

Currently (as of 2007), the ETL market is a multi-million market. All the major DBMS vendors provide ETL solutions shipped with their data warehouse suites; e.g., IBM with WebSphere Datastage, Microsoft with SQL Server Integration Services, Oracle with Warehouse Builder, and so on. In addition, individual vendors provide a large variety of ETL tools; e.g., Informatica

with PowerCenter, while some open source tools exist as well; e.g., Talend.

Apart from the traditional data warehouse environment, the ETL technology is useful in other modern applications too. Such applications are the mashups applications, which integrate data dynamically obtained via web-service invocations to more than one sources into an integrated experience; e.g., Yahoo Pipes (http://pipes.yahoo.com/), Google Maps (http://maps.google.com/), IBM Damia (http://services.alphaworks.ibm.com/damia/), and Microsoft Popfly (http://www.popfly.com/). The core functionality of those applications is `pure' ETL.

Therefore, the prediction for the future of ETL is very positive. In the same sense, several studies (Giga Information Group, 2002; Gartner, 2003) and a recent chapter (Simitsis et al., 2006) account the ETL as a remaining challenge and pinpoint several topics for future work:

- Integration of ETL with: XML adapters, EAI (Enterprise Application Integration) tools (e.g., MQ-Series), customized data quality tools, and the move towards parallel processing of the ETL workflows,
- Active or real-time ETL (Adzic & Fiore, 2003; Polyzotis et al., 2007), meaning the need to refresh the warehouse with as fresh data as possible (ideally, on-line), and
- Extension of the ETL mechanisms for non-traditional data, like XML/HTML, spatial and biomedical data.

CONCLUSION

ETL tools are pieces of software responsible for the extraction of data from several sources, their cleansing, customization, and insertion into a data warehouse. In all the phases of an ETL process (extraction and exportation, transformation and cleaning, and loading), individual issues arise, and, along with the problems and constraints that concern the overall ETL process, make its lifecycle a very troublesome task. The key factors underlying the main problems of ETL workflows are: (a) vastness of the data volumes; (b) quality problems, since data are not always clean and have to be cleansed; (c) performance, since the whole process has to take place within a specific time window; and (d) evolution of the sources and the data warehouse can eventually lead even to daily maintenance operations. The state of the art in the field of both research and commercial ETL tools indicate significant progress. Still, a lot of work remains to be done, as several issues are technologically open and present interesting research topics in the field of data integration in data warehouse environments.

REFERENCES

Adzic, J., and Fiore, V. (2003). Data Warehouse Population Platform. *In Proceedings of 5th International Workshop on the Design and Management of Data Warehouses (DMDW)*, Berlin, Germany.

Arktos II. (2004). A Framework for Modeling and Managing ETL Processes. Available at: http://www.dblab.ece.ntua.gr/~asimi

Borkar, V., Deshmuk, K., and Sarawagi, S. (2000). Automatically Extracting Structure form Free Text Addresses. *Bulletin of the Technical Committee on Data Engineering*, 23(4).

Calì, A. et al., (2003). IBIS: Semantic data integration at work. *In Proceedings of the 15th CAiSE*, Vol. 2681 of Lecture Notes in Computer Science, pages 79-94, Springer.

Galhardas, H., Florescu, D., Shasha, D., and Simon, E. (2000). Ajax: An Extensible Data Cleaning Tool. *In Proceedings ACM SIGMOD International Conference On the Management of Data*, page 590, Dallas, Texas.

Gartner. (2003). ETL Magic Quadrant Update: Market Pressure Increases. Available at http://www.gartner.com/reprints/informatica/112769.html

Giga Information Group. (2002). Market Overview Update: ETL. Technical Report RPA-032002-00021.

Inmon, W.-H. (1996). Building the Data Warehouse. 2nd edition. John Wiley & Sons, Inc., New York.

Jarke, M., Lenzerini, M., Vassiliou, Y., and Vassiliadis, P. (eds.) (2000). Fundamentals of Data Warehouses. 1st edition. Springer-Verlag.

Kimball, R., Reeves, L., Ross, M., and Thornthwaite, W. (1998). The Data Warehouse Lifecycle Toolkit: Expert

Methods for Designing, Developing, and Deploying Data Warehouses. John Wiley & Sons.

Labio, W., and Garcia-Molina, H. (1996). Efficient Snapshot Differential Algorithms for Data Warehousing. *In Proceedings of 22nd International Conference on Very Large Data Bases (VLDB)*, pages 63-74, Bombay, India.

Lenzerini, M. (2002). Data Integration: A Theoretical Perspective. *In Proceedings of 21st Symposium on Principles of Database Systems (PODS)*, pages 233-246, Wisconsin, USA.

Luján-Mora, S., Vassiliadis, P., Trujillo, J. (2004). Data mapping diagrams for data warehouse design with UML. *In Proceedings of the 23rd International Conference on Conceptual Modeling (ER)*, pages 191-204.

Monge, A. (2000). Matching Algorithms Within a Duplicate Detection System. *Bulletin of the Technical Committee on Data Engineering*, 23(4).

Polyzotis, N., Skiadopoulos, S., Vassiliadis, P., Simitsis, A., Frantzell, N.-E. (2007). Supporting Streaming Updates in an Active Data Warehouse. *In Proceedings of the 23rd IEEE International Conference on Data Engineering (ICDE)*, Istanbul, Turkey.

Rahm, E., and Do, H.Hai (2000). Data Cleaning: Problems and Current Approaches. *Bulletin of the Technical Committee on Data Engineering*, 23(4).

Raman, V., and Hellerstein, J. (2001). Potter's Wheel: An Interactive Data Cleaning System. *In Proceedings of 27th International Conference on Very Large Data Bases (VLDB)*, pages 381-390, Roma, Italy.

Rundensteiner, E. (editor) (1999). Special Issue on Data Transformations. Bulletin of the Technical Committee on Data Engineering, 22(1).

Sarawagi, S. (2000). Special Issue on Data Cleaning. Bulletin of the Technical Committee on Data Engineering, 23(4).

Scalzo, B., (2003). Oracle DBA Guide to Data Warehousing and Star Schemas. Prentice Hall PTR.

Skoutas D., Simitsis, A. (2006). Designing ETL processes using semantic web technologies. *In Proceedings of the ACM 9th International Workshop on Data Warehousing and OLAP (DOLAP)*, pages 67-74, Arlington, USA.

Simitsis, A. (2003). List of ETL Tools. Available at: http://www.dbnet.ece.ntua.gr/~asimi/ETLTools.htm

Simitsis, A. (2004). Modeling and Managing Extraction-Transformation-Loading (ETL) Processes in Data Warehouse Environments. PhD Thesis. National Technical University of Athens, Greece.

Simitsis, A., Vassiliadis, P., Sellis, T.K. (2005). State-space optimization of ETL workflows. In IEEE Transactions on Knowledge and Data Engineering (TKDE), 17(10):1404-1419.

Simitsis, A., Vassiliadis, P., Skiadopoulos, S., Sellis, T. (2006). Data Warehouse Refreshment. Data Warehouses and OLAP: Concepts, Architectures and Solutions. R. Wrembel and C. Koncilia Eds., ISBN 1-59904-364-5, IRM Press.

Theodoratos, D,. Ligoudistianos, S., Sellis, T. (2001). View selection for designing the global data warehouse. *Data & Knowledge Engineering*, 39(3), 219-240.

Trujillo J., Luján-Mora, S. (2003). A UML based approach for modeling ETL processes in data warehouses. *In Proceedings of the 22nd International Conference on Conceptual Modeling (ER)*, pages 307-320.

Tziovara, V., Vassiliadis, P., Simitsis, A. (2007). Deciding the Physical Implementation of ETL Workflows. *In Proceedings of the 10th ACM International Workshop on Data Warehousing and OLAP (DOLAP)*, Lisbon, Portugal.

Vassiliadis, P. (2000). Gulliver in the land of data warehousing: practical experiences and observations of a researcher. *In Proceedings of 2nd International Workshop on Design and Management of Data Warehouses (DMDW)*, pages 12.1–12.16, Sweden.

Vassiliadis, P., Karagiannis, A., Tziovara, V., Simitsis, A. (2007). Towards a Benchmark for ETL Workflows. *In Proceedings of the 5th International Workshop on Quality in Databases (QDB) at VLDB*, Vienna, Austria.

Vassiliadis, P., Simitsis, A., Skiadopoulos, S. (2002). Conceptual modeling for ETL processes. *In Proceedings of the ACM 5th International Workshop on Data Warehousing and OLAP (DOLAP)*, pages 14-21, McLean, USA.

Wikipedia. (2007). ETL tools. Availaible at http://en.wikipedia.org/wiki/Category:ETL tools.

KEY TERMS

Active Data Warehouse: A data warehouse that continuously is updated. The population of an active data warehouse is realized in a streaming mode. Its main characteristic is the freshness of data; the moment a change occurs in the source site, it should be propagated to the data warehouse immediately. (Similar term: Real-time Data Warehouse.)

Data Mart: A logical subset of the complete data warehouse. We often view the data mart as the restriction of the data warehouse to a single business process or to a group of related business processes targeted towards a particular business group.

Data Staging Area (DSA): An auxiliary area of volatile data employed for the purpose of data transformation, reconciliation and cleaning before the final loading of the data warehouse.

Data Warehouse: A Data Warehouse is a subject-oriented, integrated, time-variant, non-volatile collection of data used to support the strategic decision-making process for the enterprise. It is the central point of data integration for business intelligence and is the source of data for the data marts, delivering a common view of enterprise data (Inmon, 1996).

ETL: Extract, transform, and load (ETL) are data warehousing operations, which extract data from outside sources, transform it to fit business needs, and ultimately load it into the data warehouse. ETL is an important part of data warehousing, as it is the way data actually gets loaded into the warehouse.

On-Line Analytical Processing (OLAP): The general activity of querying and presenting text and number data from data warehouses, as well as a specifically dimensional style of querying and presenting that is exemplified by a number of "OLAP vendors".

Source System: The physical machine(-s) on which raw information is stored. It is an operational system whose semantics capture the transactions of the business.

Target System: The physical machine on which the warehouse's data is organized and is stored for direct querying by end users, report writers, and other applications.

Data Warehouse Performance

Beixin (Betsy) Lin
Montclair State University, USA

Yu Hong
Colgate-Palmolive Company, USA

Zu-Hsu Lee
Montclair State University, USA

INTRODUCTION

A data warehouse is a large electronic repository of information that is generated and updated in a structured manner by an enterprise over time to aid business intelligence and to support decision making. Data stored in a data warehouse is non-volatile and time variant and is organized by subjects in a manner to support decision making (Inmon et al., 2001). Data warehousing has been increasingly adopted by enterprises as the backbone technology for business intelligence reporting and query performance has become the key to the successful implementation of data warehouses. According to a survey of 358 businesses on reporting and end-user query tools, conducted by Appfluent Technology, data warehouse performance significantly affects the Return on Investment (ROI) on Business Intelligence (BI) systems and directly impacts the bottom line of the systems (Appfluent Technology, 2002). Even though in some circumstances it is very difficult to measure the benefits of BI projects in terms of ROI or dollar figures, management teams are still eager to have a "single version of the truth," better information for strategic and tactical decision making, and more efficient business processes by using BI solutions (Eckerson, 2003).

Dramatic increases in data volumes over time and the mixed quality of data can adversely affect the performance of a data warehouse. Some data may become outdated over time and can be mixed with data that are still valid for decision making. In addition, data are often collected to meet potential requirements, but may never be used. Data warehouses also contain external data (e.g. demographic, psychographic, etc.) to support a variety of predictive data mining activities. All these factors contribute to the massive growth of data volume. As a result, even a simple query may become burdensome to process and cause overflowing system indices (Inmon et al., 1998). Thus, exploring the techniques of performance tuning becomes an important subject in data warehouse management.

BACKGROUND

There are inherent differences between a traditional database system and a data warehouse system, though to a certain extent, all databases are similarly designed to serve a basic administrative purpose, e.g., to deliver a quick response to transactional data processes such as entry, update, query and retrieval. For many conventional databases, this objective has been achieved by online transactional processing (OLTP) systems (e.g. Oracle Corp, 2004; Winter and Auerbach, 2004). In contrast, data warehouses deal with a huge volume of data that are more historical in nature. Moreover, data warehouse designs are strongly organized for decision making by subject matter rather than by defined access or system privileges. As a result, a dimension model is usually adopted in a data warehouse to meet these needs, whereas an Entity-Relationship model is commonly used in an OLTP system. Due to these differences, an OLTP query usually requires much shorter processing time than a data warehouse query (Raden, 2003). Performance enhancement techniques are, therefore, especially critical in the arena of data warehousing.

Despite the differences, these two types of database systems share some common characteristics. Some techniques used in a data warehouse to achieve a better performance are similar to those used in OLTP, while some are only developed in relation to data warehousing. For example, as in an OLTP system, an index is also used in a data warehouse system, though a data

warehouse might have different kinds of indexing mechanisms based on its granularity. Partitioning is a technique which can be used in data warehouse systems as well (Silberstein et al., 2003).

On the other hand, some techniques are developed specifically to improve the performance of data warehouses. For example, aggregates can be built to provide a quick response time for summary information (e.g. Eacrett, 2003; Silberstein, 2003). Query parallelism can be implemented to speed up the query when data are queried from several tables (Silberstein et al., 2003). Caching and query statistics are unique for data warehouses since the statistics will help to build a smart cache for better performance. Also, precalculated reports are useful to certain groups of users who are only interested in seeing static reports (Eacrett, 2003). Periodic data compression and archiving helps to cleanse the data warehouse environment. Keeping only the necessary data online will allow faster access (e.g. Kimball, 1996).

MAIN THRUST

As discussed earlier, performance issues play a crucial role in a data warehouse environment. This chapter describes ways to design, build, and manage data warehouses for optimum performance. The techniques of tuning and refining the data warehouse discussed below have been developed in recent years to reduce operating and maintenance costs and to substantially improve the performance of new and existing data warehouses.

Performance Optimization at the Data Model Design Stage

Adopting a good data model design is a proactive way to enhance future performance. In the data warehouse design phase, the following factors should be taken into consideration.

- **Granularity:** Granularity is the main issue that needs to be investigated carefully before the data warehouse is built. For example, does the report need the data at the level of store keeping units (SKUs), or just at the brand level? These are size questions that should be asked of business users before designing the model. Since a data warehouse is a decision support system, rather than a transactional system, the level of detail required is usually not as deep as the latter. For instance, a data warehouse does not need data at the document level such as sales orders, purchase orders, which are usually needed in a transactional system. In such a case, data should be summarized before they are loaded into the system. Defining the data that are needed – no more and no less – will determine the performance in the future. In some cases, the Operational Data Stores (ODS) will be a good place to store the most detailed granular level data and those data can be provided on the jump query basis.
- **Cardinality:** Cardinality means the number of possible entries of the table. By collecting business requirements, the cardinality of the table can be decided. Given a table's cardinality, an appropriate indexing method can then be chosen.
- **Dimensional Models:** Most data warehouse designs use dimensional models, such as Star-Schema, Snow-Flake, and Star-Flake. A star-schema is a dimensional model with fully denormalized hierarchies, whereas a snowflake schema is a dimensional model with fully normalized hierarchies. A star-flake schema represents a combination of a star schema and a snow-flake schema (e.g. Moody and Kortink, 2003). Data warehouse architects should consider the pros and cons of each dimensional model before making a choice.

Aggregates

Aggregates are the subsets of the fact table data (Eacrett, 2003). The data from the fact table are summarized into aggregates and stored physically in a different table than the fact table. Aggregates can significantly increase the performance of the OLAP query since the query will read fewer data from the aggregates than from the fact table. Database read time is the major factor in query execution time. Being able to reduce the database read time will help the query performance a great deal since fewer data are being read. However, the disadvantage of using aggregates is its loading performance. The data loaded into the fact table have to be rolled up to the aggregates, which means any newly updated records will have to be updated in the aggregates as well to make the data in the aggregates

consistent with those in the fact table. Keeping the data as current as possible has presented a real challenge to data warehousing (e.g. Bruckner and Tjoa, 2002). The ratio of the database records transferred to the database records read is a good indicator of whether or not to use the aggregate technique. In practice, if the ratio is 1/10 or less, building aggregates will definitely help performance (Silberstein, 2003).

Database Partitioning

Logical partitioning means using year, planned/actual data, and business regions as criteria to partition the database into smaller data sets. After logical partitioning, a database view is created to include all the partitioned tables. In this case, no extra storage is needed and each partitioned table will be smaller to accelerate the query (Silberstein et al., 2003). Take a multinational company as an example. It is better to put the data from different countries into different data targets, such as cubes or data marts, than to put the data from all the countries into one data target. By logical partitioning (splitting the data into smaller cubes), query can read the smaller cubes instead of large cubes and several parallel processes can read the small cubes at the same time. Another benefit of logical partitioning is that each partitioned cube is less complex to load and easier to perform the administration. Physical partitioning can also reach the same goal as logical partitioning. Physical partitioning means that the database table is cut into smaller chunks of data. The partitioning is transparent to the user. The partitioning will allow parallel processing of the query and each parallel process will read a smaller set of data separately.

Query Parallelism

In business intelligence reporting, the query is usually very complex and it might require the OLAP engine to read the data from different sources. In such case, the technique of query parallelism will significantly improve the query performance. Query parallelism is an approach to split the query into smaller sub-queries and to allow parallel processing of sub-queries. By using this approach, each sub-process takes a shorter time and reduces the risk of system hogging if a single long-running process is used. In contrast, sequential processing definitely requires a longer time to process (Silberstein et al., 2003).

Database Indexing

In a relational database, indexing is a well known technique for reducing database read time. By the same token, in a data warehouse dimensional model, the use of indices in the fact table, dimension table, and master data table will improve the database read time.

- **The Fact Table Index:** By default the fact table will have the primary index on all the dimension keys. However, a secondary index can also be built to fit a different query design (e.g. McDonald et al., 2002). Unlike a primary index, which includes all the dimension keys, the secondary index can be built to include only some dimension keys to improve the performance. By having the right index, the query read time can be dramatically reduced.
- **The Dimension Table Index:** In a dimensional model, the size of the dimension table is the deciding factor affecting query performance. Thus, the index of the dimension table is important to decrease the master data read time, and thus to improve filtering and drill down. Depending on the cardinality of the dimension table, different index methods will be adopted to build the index. For a low cardinality dimension table, Bit-Map index is usually adopted. In contrast, for a high cardinality dimension table, the B-Tree index should be used (e.g. McDonald et al., 2002).
- **The Master Data Table Index:** Since the data warehouse commonly uses dimensional models, the query SQL plan always starts from reading the master data table. Using indices on the master data table will significantly enhance the query performance.

Caching Technology

Caching technology will play a critical role in query performance as the memory size and speed of CPUs increase. After a query runs once, the result is stored in the memory cache. Subsequently, similar query runs will get the data directly from the memory rather than by accessing the database again. In an enterprise server, a certain block of memory is allocated for the caching use. The query results and the navigation status can then be stored in a highly compressed format in that memory block. For example, a background user can

be set up during the off-peak time to mimic the way a real user runs the query. By doing this, a cache is generated that can be used by real business users.

Pre-Calculated Report

Pre-calculation is one of the techniques where the administrator can distribute the workload to off-peak hours and have the result sets ready for faster access (Eacrett, 2003). There are several benefits of using the pre-calculated reports. The user will have faster response time since calculation takes place on the fly. Also, the system workload is balanced and shifted to off-peak hours. Lastly, the reports can be available offline.

Use Statistics to Further Tune Up the System

In a real-world data warehouse system, the statistics data of the OLAP are collected by the system. The statistics provide such information as what the most used queries are and how the data are selected. The statistics can help further tune the system. For example, examining the descriptive statistics of the queries will reveal the most common used drill-down dimensions as well as the combinations of the dimensions. Also, the OLAP statistics will also indicate what the major time component is out of the total query time. It could be database read time or OLAP calculation time. Based on the data, one can build aggregates or offline reports to increase the query performance.

Data Compression

Over time, the dimension table might contain some unnecessary entries or redundant data. For example, when some data has been deleted from the fact table, the corresponding dimension keys will not be used any more. These keys need to be deleted from the dimension table in order to have a faster query time since the query execution plan always start from the dimension table. A smaller dimension table will certainly help the performance.

The data in the fact table need to be compressed as well. From time to time, some entries in the fact table might contain just all zeros and need to be removed. Also, entries with the same dimension key value should be compressed to reduce the size of the fact table.

Kimball (1996) pointed out that data compression might not be a high priority for an OLTP system since the compression will slow down the transaction process. However, compression can improve data warehouse performance significantly. Therefore, in a dimensional model, the data in the dimension table and fact table need to be compressed periodically.

Data Archiving

In the real data warehouse environment, data are being loaded daily or weekly. The volume of the production data will become huge over time. More production data will definitely lengthen the query run time. Sometimes, there are too much unnecessary data sitting in the system. These data may be out-dated and not important to the analysis. In such cases, the data need to be periodically archived to offline storage so that they can be removed from the production system (e.g. Uhle, 2003). For example, three years of production data are usually sufficient for a decision support system. Archived data can still be pulled back to the system if such need arises

FUTURE TRENDS

Although data warehouse applications have been evolving to a relatively mature stage, several challenges remain in developing a high performing data warehouse environment. The major challenges are:

- How to optimize the access to volumes of data based on system workload and user expectations (e.g. Inmon et al., 1998)? Corporate information systems rely more and more on data warehouses that provide one entry point for access to all information for all users. Given this, and because of varying utilization demands, the system should be tuned to reduce the system workload during peak user periods.
- How to design a better OLAP statistics recording system to measure the performance of the data warehouse? As mentioned earlier, query OLAP statistics data is important to identify the trend and the utilization of the queries by different users. Based on the statistics results, several techniques could be used to tune up the performance, such as aggregates, indices and caching.

- How to maintain data integrity and keep the good data warehouse performance at the same time? Information quality and ownership within the data warehouse play an important role in the delivery of accurate processed data results (Ma et al., 2000). Data validation is thus a key step towards improving data reliability and quality. Accelerating the validation process presents a challenge to the performance of the data warehouse.

CONCLUSION

Information technologies have been evolving quickly in recent decades. Engineers have made great strides in developing and improving databases and data warehousing in many aspects, such as hardware; operating systems; user interface and data extraction tools; data mining tools, security, and database structure and management systems.

Many success stories in developing data warehousing applications have been published (Beitler and Leary, 1997; Grim and Thorton, 1997). An undertaking of this scale – creating an effective data warehouse, however, is costly and not risk-free. Weak sponsorship and management support, insufficient funding, inadequate user involvement, and organizational politics have been found to be common factors contributing to failure (Watson et al., 1999). Therefore, data warehouse architects need to first clearly identify the information useful for the specific business process and goals of the organization. The next step is to identify important factors that impact performance of data warehousing implementation in the organization. Finally all the business objectives and factors will be incorporated into the design of the data warehouse system and data models, in order to achieve high-performing data warehousing and business intelligence.

REFERENCES

Appfluent Technology. (December 2, 2002). A study on reporting and business intelligence application usage. The Internet <http://searchsap.techtarget.com/whitepaperPage/0,293857,sid21_gci866993,00.html >

Beitler, S. S., & Leary, R. (1997). Sears' EPIC transformation: converting from mainframe legacy systems to On-Line Analytical Processing (OLAP). *Journal of Data Warehousing*, *2*, 5-16.

Bruckner, R. M., & Tjoa, A. M. (2002). Capturing delays and valid times in data warehouses--towards timely consistent analyses. *Journal of Intelligent Information Systems*, *19*(2), 169-190.

Eacrett, M. (2003). Hitchhiker's guide to SAP business information warehouse performance tuning. SAP White Paper.

Eckerson, W. (2003). BI StatShots. *Journal of Data Warehousing*, *8*(4), 64.

Grim, R., & Thorton, P. A. (1997). A customer for life: the warehouseMCI approach. *Journal of Data Warehousing*, *2*, 73-79.

Inmon, W.H., Rudin, K., Buss, C. K., & Sousa, R. (1998). *Data Warehouse Performance*. New York: John Wiley & Sons, Inc.

Imhoff, C., Sousa (2001). *Corporate Information Factory*. New York: John Wiley & Sons, Inc.

Kimball, R. (1996). *The Data Warehouse Toolkit***.** New York: John Wiley & Sons, Inc.

Ma, C., *Chou, D. C.*, & *Yen, D. C.* (2000). Data warehousing, technology assessment and management, *Industrial Management + Data Systems*, 100, 125

McDonald, K., Wilmsmeier, A., Dixon, D. C., & Inmon, W.H. (2002, August). *Mastering the SAP Business Information Warehouse.* Hoboken, NJ: John Wiley & Sons, Inc.

Moody D., & Kortink, M. A. R. (2003). From ER models to dimensional models, part II: advanced design issues, *Journal of Data Warehousing*, *8*, 20-29.

Oracle Corp. (2004). Largest transaction processing db on Unix runs oracle database. *Online Product News*, 23(2).

Peterson, S. (1994). *Stars: a pattern language for query optimized schema.* Sequent Computer Systems, Inc. White Paper.

Raden, N. (2003). Real time: get real, part II. *Intelligent Enterprise*, 6 (11), 16.

Silberstein, R. (2003). Know How Network: SAP BW Performance Monitoring with BW Statistics. *SAP White Paper*.

Silberstein, R., Eacrett, M., Mayer, O., & Lo, A. (2003). SAP BW performance tuning. *SAP White Paper*.

Uhle, R. (2003). Data aging with mySAP business intelligence. SAP White Paper.

Watson, H. Gerard, J., J.G., Gonzalez, L. E., Haywood, M. E., & Fenton, D. (1999). Data warehousing failures: case Studies and findings. *Journal of Data Warehousing*, 4, 44-55.

Winter, R. & Auerbach, K. (2004). Contents under pressure: scalability challenges for large databases. *Intelligent Enterprise*, 7 (7), 18-25.

KEY TERMS

Cache: A region of a computer's memory which stores recently or frequently accessed data so that the time of repeated access to the same data can decrease.

Granularity: The level of detail or complexity at which an information resource is described.

Indexing: In data storage and retrieval, the creation and use of a list that inventories and cross-references data. In database operations, a method to find data more efficiently by indexing on primary key fields of the database tables.

ODS (Operational Data Stores): A system with capability of continuous background update that keeps up with individual transactional changes in operational systems versus a data warehouse that applies a large load of updates on an intermittent basis.

OLTP (Online Transaction Processing): A standard, normalized database structure designed for transactions in which inserts, updates, and deletes must be fast.

OLAP (Online Analytical Processing): A category of software tools for collecting, presenting, delivering, processing and managing multidimensional data (i.e., data that has been aggregated into various categories or "dimensions") in order to provide analytical insights for business management.

Service Management: The strategic discipline for identifying, establishing, and maintaining IT services to support the organization's business goal at an appropriate cost.

SQL (Structured Query Language): A standard interactive programming language used to communicate with relational databases in order to retrieve, update, and manage data.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 318-322, copyright 2005 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

Data Warehousing and Mining in Supply Chains

Richard Mathieu
Saint Louis University, USA

Reuven R. Levary
Saint Louis University, USA

INTRODUCTION

Every finished product has gone through a series of transformations. The process begins when manufacturers purchase the raw materials that will be transformed into the components of the product. The parts are then supplied to a manufacturer, who assembles them into the finished product and ships the completed item to the consumer. The transformation process includes numerous activities (Levary, 2000). Among them are

- Designing the product
- Designing the manufacturing process
- Determining which component parts should be produced in house and which should be purchased from suppliers
- Forecasting customer demand
- Contracting with external suppliers for raw materials or component parts
- Purchasing raw materials or component parts from suppliers
- Establishing distribution channels for raw materials and component parts from suppliers to manufacturer
- Establishing of distribution channels to the suppliers of raw materials and component parts
- Establishing distribution channels from the manufacturer to the wholesalers and from wholesalers to the final customers
- Manufacturing the component parts
- Transporting the component parts to the manufacturer of the final product
- Manufacturing and assembling the final product
- Transporting the final product to the wholesalers, retailers, and final customer

Each individual activity generates various data items that must be stored, analyzed, protected, and transmitted to various units along a supply chain.

A *supply chain* can be defined as a series of activities that are involved in the transformation of raw materials into a final product, which a customer then purchases (Levary, 2000). The flow of materials, component parts, and products is moving downstream (i.e., from the initial supply sources to the end customers). The flow of information regarding the demand for the product and orders to suppliers is moving upstream, while the flow of information regarding product availability, shipment schedules, and invoices is moving downstream. For each organization in the supply chain, its customer is the subsequent organization in the supply chain, and its subcontractor is the prior organization in the chain.

BACKGROUND

Supply chain data can be characterized as either transactional or analytical (Shapiro, 2001). All new data that are acquired, processed, and compiled into reports that are transmitted to various organizations along a supply chain are deemed transactional data (Davis & Spekman, 2004). Increasingly, transactional supply chain data is processed and stored in enterprise resource planning systems, and complementary data warehouses are developed to support decision-making processes (Chen R., Chen, C., & Chang, 2003; Zeng, Chiang, & Yen, 2003). Organizations such as Home Depot, Lowe's, and Volkswagen have developed data warehouses and integrated data-mining methods that complement their supply chain management operations (Dignan, 2003a; Dignan, 2003b; Hofmann, 2004). Data that are used in descriptive and optimization models are considered

analytical data (Shapiro, 2001). Descriptive models include various forecasting models, which are used to forecast demands along supply chains, and managerial accounting models, which are used to manage activities and costs. Optimization models are used to plan resources, capacities, inventories, and product flows along supply chains.

Data collected from consumers are the core data that affect all other data items along supply chains. Information collected from the consumers at the point of sale include data items regarding the sold product (e.g., type, quantity, sale price, and time of sale) as well as information about the consumer (e.g., consumer address and method of payment). These data items are analyzed often. Data-mining techniques are employed to determine the types of items that are appealing to consumers. The items are classified according to consumers' socioeconomic backgrounds and interests, the sale price that consumers are willing to pay, and the location of the point of sale.

Data regarding the return of sold products are used to identify potential problems with the products and their uses. These data include information about product quality, consumer disappointment with the product, and legal consequences. Data-mining techniques can be used to identify patterns in returns so that retailers can better determine which type of product to order in the future and from which supplier it should be purchased. Retailers are also interested in collecting data regarding competitors' sales so that they can better promote their own product and establish a competitive advantage.

Data related to political and economic conditions in supplier countries are of interest to retailers. Data-mining techniques can be used to identify political and economic patterns in countries. Information can help retailers choose suppliers who are situated in countries where the flow of products and funds is expected to be stable for a reasonably long period of time.

Manufacturers collect data regarding a) particular products and their manufacturing process, b) suppliers, and c) the business environment. Data regarding the product and the manufacturing process include the characteristics of products and their component parts obtained from CAD/CAM systems, the quality of products and their components, and trends in theresearch and development (R & D) of relevant technologies. Data-mining techniques can be applied to identify patterns in the defects of products, their components, or the manufacturing process. Data regarding suppliers include availability of raw materials, labor costs, labor skills, technological capability, manufacturing capacity, and lead time of suppliers. Data related to qualified teleimmigrants (e.g., engineers and computer software developers) is valuable to many manufacturers. Data-mining techniques can be used to identify those teleimmigrants having unique knowledge and experience. Data regarding the business environment of manufacturers include information about competitors, potential legal consequences regarding a product or service, and both political and economic conditions in countries where the manufacturer has either facilities or business partners. Data-mining techniques can be used to identify possible liability concerning a product or service as well as trends in political and economic conditions in countries where the manufacturer has business interests.

Retailers, manufacturers, and suppliers are all interested in data regarding transportation companies. These data include transportation capacity, prices, lead time, and reliability for each mode of transportation.

MAIN THRUST

Data Aggregation in Supply Chains

Large amounts of data are being accumulated and stored by companies belonging to supply chains. Data aggregation can improve the effectiveness of using the data for operational, tactical, and strategic planning models. The concept of data aggregation in manufacturing firms is called *group technology (GT).* Nonmanufacturing firms are also aggregating data regarding products, suppliers, customers, and markets.

Group Technology

Group technology is a concept of grouping parts, resources, or data according to similar characteristics. By grouping parts according to similarities in geometry, design features, manufacturing features, materials used, and/or tooling requirements, manufacturing efficiency can be enhanced, and productivity increased. Manufacturing efficiency is enhanced by

- Performing similar activities at the same work center so that setup time can be reduced

- Avoiding duplication of effort both in the design and manufacture of parts
- Avoiding duplication of tools
- Automating information storage and retrieval (Levary, 1993)

Effective implementation of the GT concept necessitates the use of a classification and coding system. Such a system codes the various attributes that identify similarities among parts. Each part is assigned a number or alphanumeric code that uniquely identifies the part's attributes or characteristics. A part's code must include both design and manufacturing attributes.

A classification and coding system must provide an effective way of grouping parts into part families. All parts in a given part family are similar in some aspect of design or manufacture. A part may belong to more than one family.

A part code is typically composed of a large number of characters that allow for identification of all part attributes. The larger the number of attributes included in a part code, the more difficult the establishment of standard procedures for classifying and coding. Although numerous methods of classification and coding have been developed, none has emerged as the standard method. Because different manufacturers have different requirements regarding the type and composition of parts' codes, customized methods of classification and coding are generally required. Some of the better known classification and coding methods are listed by Groover (1987).

After a code is established for each part, the parts are grouped according to similarities and are assigned to part families. Each part family is designed to enhance manufacturing efficiency in a particular way. The information regarding each part is arranged according to part families in a GT database. The GT database is designed in such a way that users can efficiently retrieve desired information by using the appropriate code.

Consider part families that are based on similarities of design features. A GT database enables design engineers to search for existing part designs that have characteristics similar to those of a new part that is to be designed. The search begins when the design engineer describes the main characteristics of the needed part with the help of a partial code. The computer then searches the GT database for all the items with the same code. The results of the search are listed on the computer screen, and the designer can then select or modify an existing part design after reviewing its specifications. Selected designs can easily be retrieved. When design modifications are needed, the file of the selected part is transferred to a CAD system. Such a system enables the design engineer to effectively modify the part's characteristics in a short period of time. In this way, efforts are not duplicated when designing parts.

The creation of a GT database helps reduce redundancy in the purchasing of parts as well. The database enables manufacturers to identify similar parts produced by different companies. It also helps manufacturers to identify components that can serve more than a single function. In such ways, GT enables manufacturers to reduce both the number of parts and the number of suppliers. Manufacturers that can purchase large quantities of a few items rather than small quantities of many items are able to take advantage of quantity discounts.

Aggregation of Data Regarding Retailing Products

Retailers may carry thousands of products in their stores. To effectively manage the logistics of so many products, product aggregation is highly desirable. Products are aggregated into families that have some similar characteristics. Examples of product families include the following:

- Products belonging to the same supplier
- Products requiring special handling, transportation, or storage
- Products intended to be used or consumed by a specific group of customers
- Products intended to be used or consumed in a specific season of the year
- Volume and speed of product movement
- The methods of the transportation of the products from the suppliers to the retailers
- The geographical location of suppliers
- Method of transaction handling with suppliers; for example, EDI, Internet, off-line

As in the case of GT, retailing products may belong to more than one family.

Aggregation of Data Regarding Customers of Finished Products

To effectively market finished products to customers, it is helpful to aggregate customers with similar characteristics into families. Examples of customers of finished product families include

- Customers residing in a specific geographical region
- Customers belonging to a specific socioeconomic group
- Customers belonging to a specific age group
- Customers having certain levels of education
- Customers having similar product preferences
- Customers of the same gender
- Customers with the same household size

Similar to both GT and retailing products, customers of finished products may belong to more than one family.

FUTURE TRENDS

Supply Chain Decision Databases

The enterprise database systems that support supply chain management are repositories for large amounts of transaction-based data. These systems are said to be data rich but information poor. The tremendous amount of data that are collected and stored in large, distributed database systems has far exceeded the human ability for comprehension without analytic tools. Shapiro estimates that 80% of the data in a transactional database that supports supply chain management is irrelevant to decision making and that data aggregations and other analyses are needed to transform the other 20% into useful information (2001). Data warehousing and online analytical processing (OLAP) technologies combined with tools for data mining and knowledge discovery have allowed the creation of systems to support organizational decision making.

The supply chain management (SCM) data warehouse must maintain a significant amount of data for decision making. Historical and current data are required from supply chain partners and from various functional areas within the firm in order to support decision making in regard to planning, sourcing, production, and product delivery. Supply chains are dynamic in nature. In a supply chain environment, it may be desirable to learn from an archived history of temporal data that often contains some information that is less than optimal. In particular, SCM environments are typically characterized by variable changes in product demand, supply levels, product attributes, machine characteristics, and production plans. As these characteristics change over time, so does the data in the data warehouses that support SCM decision making. We should note that Kimball and Ross (2002) use a *supply value chain* and a *demand supply chain* as the framework for developing the data model for all business data warehouses.

The data warehouses provide the foundation for decision support systems (DSS) for supply chain management. Analytical tools (simulation, optimization, & data mining) and presentation tools (geographic information systems and graphical user interface displays) are coupled with the input data provided by the data warehouse (Marakas, 2003). Simchi-Levi, D., Kaminsky, and Simchi-Levi, E. (2000) describe three DSS examples: logistics network design, supply chain planning and vehicle routing, and scheduling. Each DSS requires different data elements, has specific goals and constraints, and utilizes special graphical user interface (GUI) tools.

The Role of Radio Frequency Identification (RFID) in Supply Chains Data Warehousing

The emerging RFID technology will generate large amounts of data that need to be warehoused and mined. *Radio frequency identification (RFID)* is a wireless technology that identifies objects without having either contact or sight of them. RFID tags can be read despite environmentally difficult conditions such as fog, ice, snow, paint, and widely fluctuating temperatures. Optically read technologies, such as bar codes, cannot be used in such environments. RFID can also identify objects that are moving.

Passive RFID tags have no external power source. Rather, they have operating power generated from a reader device. The passive RFID tags are very small and inexpensive. Further, they have a virtually unlimited operational life. The characteristics of these passive RFID tags make them ideal for tracking materials through supply chains. Wal-Mart has required manufacturers,

suppliers, distributors, and carriers to incorporate RFID tags into both products and operations. Other large retailers are following Wal-Mart's lead in requesting RFID tags to be installed in goods along their supply chain. The tags follow products from the point of manufacture to the store shelf. RFID technology will significantly increase the effectiveness of tracking materials along supply chains and will also substantially reduce the loss that retailers accrue from thefts. Nonetheless, civil liberty organizations are trying to stop RFID tagging of consumer goods, because this technology has the potential of affecting consumer privacy. RFID tags can be hidden inside objects without customer knowledge. So RFID tagging would make it possible for individuals to read the tags without the consumers even having knowledge of the tags' existence.

Sun Microsystems has designed RFID technology to reduce or eliminate drug counterfeiting in pharmaceutical supply chains (Jaques, 2004). This technology will make the copying of drugs extremely difficult and unprofitable. Delta Air Lines has successfully used RFID tags to track pieces of luggage from check-in to planes (Brewin, 2003). The luggage-tracking success rate of RFID was much better than that provided by bar code scanners.

Active RFID tags, unlike passive tags, have an internal battery. The tags have the ability to be rewritten and/or modified. The read/write capability of active RFID tags is useful in interactive applications such as tracking work in process or maintenance processes. Active RFID tags are larger and more expensive than passive RFID tags. Both the passive and active tags have a large, diverse spectrum of applications and have become the standard technologies for automated identification, data collection, and tracking. A vast amount of data will be recorded by RFID tags. The storage and analysis of this data will pose new challenges to the design, management, and maintenance of databases as well as to the development of data-mining techniques.

CONCLUSION

A large amount of data is likely to be gathered from the many activities along supply chains. This data must be warehoused and mined to identify patterns that can lead to better management and control of supply chains. The more RFID tags installed along supply chains, the easier data collection becomes. As the tags become more popular, the data collected by them will grow significantly. The increased popularity of the tags will bring with it new possibilities for data analysis as well as new warehousing and mining challenges.

REFERENCES

Brewin, B. (2003, December). Delta says radio frequency ID devices pass first bag. *Computer World, 7*. Retrieved March 29, 2005 from *www.computerworld.com/mobiletopics/mobile/technology/story/0,10801,88446, 00/*

Chen, R., Chen, C., & Chang, C. (2003). A Web-based ERP data mining system for decision making. *International Journal of Computer Applications in Technology, 17*(3), 156-158.

Davis, E. W., & Spekman, R. E. (2004). *Extended enterprise.* Upper Saddle River, NJ: Prentice Hall.

Dignan, L. (2003a, June 16). Data depot. *Baseline Magazine*.

Dignan, L. (2003b, June 16). Lowe's big plan. *Baseline Magazine*.

Groover, M. P. (1987). *Automation, production systems, and computer-integrated manufacturing*. Englewood Cliffs, NJ: Prentice-Hall.

Hofmann, M. (2004, March). Best practices: VW revs its B2B engine. *Optimize Magazine*, 22-30.

Jaques, R. (2004, February 20). Sun pushes RFID drug technology. *E-business/Technology/News*. Retrieved March 29, 2005, from *www.vnunet.com/News/1152921*

Kimball, R., & Ross, M. (2002). *The data warehouse toolkit* (2nd ed.). New York: Wiley.

Levary, R. R. (1993). Group technology for enhancing the efficiency of engineering activities. *European Journal of Engineering Education, 18*(3), 277-283.

Levary, R. R. (2000, May–June). Better supply chains through information technology. *Industrial Management, 42*, 24-30.

Marakas, G. M. (2003). *Modern data warehousing, mining and visualization.* Upper Saddle River, NJ: Prentice-Hall.

Shapiro, J. F. (2001). *Modeling the supply chain.* Pacific Grove, CA: Duxbury.

Simchi-Levi, D., Kaminsky, P., & Simchi-Levi, E. (2000). *Designing and managing the supply chain*. Boston, MA: McGraw-Hill.

Zeng, Y., Chiang, R., & Yen, D. (2003). Enterprise integration with advanced information technologies: ERP and data warehousing. *Information Management and Computer Security, 11*(3), 115-122.

KEY TERMS

Analytical Data: All data that are obtained from optimization, forecasting, and decision support models.

Computer Aided Design (CAD): An interactive computer graphics system used for engineering design.

Computer Aided Manufacturing (CAM): The use of computers to improve both the effectiveness and efficiency of manufacturing activities.

Decision Support System (DSS): Computer-based systems designed to assist in managing activities and information in organizations.

Graphical User Interface (GUI): A software interface that relies on icons, bars, buttons, boxes, and other images to initiate computer-based tasks for users.

Group Technology (GP): The concept of grouping parts, resources, or data according to similar characteristics.

Online Analytical Processing (OLAP): A form of data presentation in which data are summarized, aggregated, deaggregated, and viewed in the frame of a table or cube.

Supply Chain: A series of activities that are involved in the transformation of raw materials into a final product that is purchased by a customer.

Transactional Data: All data that are acquired, processed, and compiled into reports, which are transmitted to various organizations along a supply chain.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 523-528, copyright 2005 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

Data Warehousing for Association Mining

Yuefeng Li
Queensland University of Technology, Australia

INTRODUCTION

With the phenomenal growth of electronic data and information, there are many demands for developments of efficient and effective systems (tools) to address the issue of performing data mining tasks on data warehouses or multidimensional databases. Association rules describe associations between itemsets (i.e., sets of data items) (or granules). Association mining (or called association rule mining) finds interesting or useful association rules in databases, which is the crucial technique for the development of data mining. Association mining can be used in many application areas, for example, the discovery of associations between customers' locations and shopping behaviours in market basket analysis.

Association mining includes two phases. The first phase is called pattern mining that is the discovery of frequent patterns. The second phase is called rule generation that is the discovery of the interesting and useful association rules in the discovered patterns. The first phase, however, often takes a long time to find all frequent patterns that also include much noise as well (Pei and Han, 2002). The second phase is also a time consuming activity (Han and Kamber, 2000) and can generate many redundant rules (Zaki, 2004) (Xu and Li, 2007). To reduce search spaces, user constraint-based techniques attempt to find knowledge that meet some sorts of constraints. There are two interesting concepts that have been used in user constraint-based techniques: meta-rules (Han and Kamber, 2000) and granule mining (Li et al., 2006).

The aim of this chapter is to present the latest research results about data warehousing techniques that can be used for improving the performance of association mining. The chapter will introduce two important approaches based on user constraint-based techniques. The first approach requests users to inputs their meta-rules that describe their desires for certain data dimensions. It then creates data cubes based these meta-rules and then provides interesting association rules.

The second approach firstly requests users to provide condition and decision attributes that used to describe the antecedent and consequence of rules, respectively. It then finds all possible data granules based condition attributes and decision attributes. It also creates a multi-tier structure to store the associations between granules, and association mappings to provide interesting rules.

BACKGROUND

Data warehouse mainly aims to make data easily accessible, present data consistently and be adaptive and resilient to change (Kimball and Ross, 2002). A data warehouse is an application that contains a collection of data, which is subject-oriented, integrated, non-volatile and time-variant, supporting management's decisions (Inmon, 2005). Data warehousing focuses on constructing and using data warehouses. The construction includes data cleaning, data integration and data consolidation. After these steps, a collection of data in a specific form can be stored in a data warehouse.

Data warehouses can also provide clean, integrated and complete data to improve the process of data mining (Han and Kamber, 2000). Han and Kamber also defined different levels of the integration of data mining and data warehouse. At the loosest level the data warehouse only acts as a normal data source of data mining. While at the tightest level both the data warehouse and data mining are sub-components that cooperate with each other. In a data mining oriented data warehouse, the data warehouse not only cleans and integrates data, but also tailors data to meet user constraints for knowledge discovery in databases. Thus, data mining can return what users want in order to improve the quality of discovered knowledge.

It is painful when we review the two steps in association mining: both take a long time and contain uncertain information for determining useful knowledge. Data mining oriented data warehousing is a promising direction for solving this problem. It refers

to constructing systems, in which both the data mining and data warehouse are a sub-component cooperating with each other. Using these systems, the data warehouse not only cleans and integrates data, but tailors data to fit the requirements of data mining. Thus, data mining becomes more efficient and accurate. In this chapter we discuss how data warehousing techniques are useful for association mining.

MAIN FOCUS

Based on the above introduction, we understand that data warehousing techniques can be helpful for improving the quality of data mining. We will focus on two areas in this chapter: mining patterns and meta-rules through data cubes and mining granules and decision rules through multi-tier structures.

Mining Patterns and Meta Rules through Data Cubes

There are many studies that discuss the research issue of performing data mining tasks on a data warehouse. The main research point is that pattern mining and rule generation can be used together with OLAP (On-Line Analytical Processing) to find interesting knowledge from data cubes (Han and Kamber, 2000) (Imielinski et al., 2002) (Messaoud et al., 2006).

A data cube consists of a set of data dimensions and a set of measures. Each dimension usually has a set of hierarchical levels. For example, a product dimension may have three hierarchal levels: All, Family and Article, where Article could be a set of attributes (e.g., {iTwin, iPower, DV-400, EN-700, aStar, aDream}), Family could be {Desktop, Laptop, MP3} and All is the total aggregation level. The frequency measure in data cube is called COUNT, which is used to evaluate the occurrences of the corresponding data values for data cube cells.

A meta-rule in multidimensional databases usually has the following format:

$$\text{“}P_1 \wedge P_2 \wedge \ldots \wedge P_m \rightarrow Q_1 \wedge Q_2 \wedge \ldots \wedge Q_n\text{”}$$

where P_i and Q_j are some predicates which can be sets of data fields in different dimension levels. The meta-rule defines the portion of the data cube to be mined.

The mining process starts to provide a meta-rule and define a minimum support and a minimum confidence. The traditional way to define the support and confidence is the use of the numbers of occurrences of data values. For example, the support can be computed according to the frequency of units of facts based on the COUNT measure. It was also recommended in the OLAP context (Messaoud et al., 2006) that users were often interested in observing facts based on summarized values of measures rather than the numbers of occurrences. Therefore, SUM measure based definitions for the support and confidence were presented in (Messaoud et al., 2006). The second step is to choose an approach (the top-down approach or bottom up approach) to produce frequent itemsets. The last step is to generate interesting association rules to meet the requirements in the meta-rule.

Mining Decision Rules through Multi-Tier Structures

User constraint-based association mining can also be implemented using granule mining, which finds interesting associations between granules in databases, where a granule is a predicate that describes common features of a set of objects (e.g., records, or transactions) for a selected set of attributes (or items).

Formally, a multidimensional database can be described as an information table (T, V^T), where T is a set of objects (records) in which each record is a sequences of items, and $V^T = \{a_1, a_2, \ldots, a_n\}$ is a set of selected items (or called attributes in decision tables) for all objects in T. Each item can be a tuple (e.g., <*name*, *cost*, *price*> is a product item).

Table I illustrates an information table, where $V^T = \{a_1, a_2, \ldots, a_7\}$, $T = \{t_1, t_2, \ldots, t_6\}$.

Given an information table, a user can classify attributes into two categories: condition attributions and decision attributes. For example, the high profit products can be condition attributes and low profit products can be decision attributes. Formally, a decision table is a tuple (T, V^T, C, D), where T is a set of objects (records) in which each record is a set of items, and $V^T = \{a_1, a_2, \ldots, a_n\}$ is a set of attributes, $C \cup D \subseteq V^T$ and $C \cap D = \varnothing$.

Objects in a decision table can also be compressed into a granule if they have the same representation (Pawlak, 2002). Table II shows a decision table of the

Table 1. An information table

Object	Items
t_1	a_1 a_2
t_2	a_3 a_4 a_6
t_3	a_3 a_4 a_5 a_6
t_4	a_3 a_4 a_5 a_6
t_5	a_1 a_2 a_6 a_7
t_6	a_1 a_2 a_6 a_7

Table 2. A decision table

Granule	**Products**							**coverset**
	High profit					*Low profit*		
	a_1	a_2	a_3	a_4	a_5	a_6	a_7	
g_1	1	1	0	0	0	0	0	$\{t_1\}$
g_2	0	0	1	1	0	1	0	$\{t_2\}$
g_3	0	0	1	1	1	1	0	$\{t_3, t_4\}$
g_4	1	1	0	0	0	1	1	$\{t_5, t_6\}$

information table in Table I, where $C = \{a_1, a_2, \ldots, a_5\}$, $D = \{a_6, a_7\}$, the set of granules is $\{g_1, g_2, g_3, g_4\}$, and *coverset* is the set of objects that are used to produce a granule.

Each granule can be viewed as a decision rule, where the antecedent is the corresponding part of condition attributes in the granule and the consequence is the corresponding part of decision attributes in the granule.

In (Li and Zhong, 2003), larger granules in decision tables were split into smaller granules and deployed into two tiers: *C-granules* (condition granules) which are granules that only use condition attributes and *D-granules* (decision granules) which are granules that only use decision attributes. Table III illustrates a 2-tier structure for the decision table in Table II, where both (A) and (B) include three small granules.

The following are advantages of using granules for knowledge discovery in databases:

1. A granule describes the feature of a set of objects, but a pattern is a part of an object;
2. The number of granules is much smaller than the numbers of patterns;
3. Decision tables can directly describe multiple values of items; and
4. It provides a user-oriented approach to determine the antecedent (also called premise) and consequence (conclusion) of association rules.

There are also several disadvantages when we discuss granules based on decision tables. The first problem is that we do not understand the relation between association rules (or patterns) and decision rules (or granules). Although decision tables can provide an efficient way to represent discovered knowledge with a small number of attributes, in cases of large number of attributes, decision tables lose their advantages because

Table 3. A 2-tier structure

Condition Granule	a_1	a_2	a_3	a_4	a_5	coverset
cg_1	1	1	0	0	0	$\{t_1, t_5, t_6\}$
cg_2	0	0	1	1	0	$\{t_2\}$
$\boldsymbol{cg_3}$	0	0	1	1	1	$\{t_1, t_4\}$

(a) C-granules

Decision Granule	a_6	a_7	coverset
dg_1	0	0	$\{t_1\}$
dg_2	1	0	$\{t_2, t_3, t_4\}$
$\boldsymbol{dg_3}$	1	1	$\{t_5, t_6\}$

(b) D-granules

they can not be used to organize granules with different sizes. They also have not provided a mechanism to define meaningless rules.

In (Li et al., 2006), a multi-tier structure was presented to overcome the above disadvantages of using granules for knowledge discovery in databases. In this structure, condition attributes can be further split into some tiers and the large multidimensional database is compressed into granules at each tier. The antecedent and consequent of an association rule are both granules, and the associations between them are described by association mappings. For example, condition attributions C can be further spit into two sets of attributes: C_i and C_j. Therefore, in the multi-tier structure, the *C-granules* tier is divided into two tiers: C_i*-granules* and C_j*-granules*.

Let cg_k be a *C-granule*, it can be represented as "$cg_{i,x} \wedge cg_{j,y}$" by using C_i*-granules* tier and C_j*-granules* tier. Decision rule "$cg_k \rightarrow dg_z$" can also have the form of

"$cg_{i,x} \wedge cg_{j,y} \rightarrow dg_z$".

Different to decision tables, we can discuss general association rules (rules with shorter premises) of decision rules. We call "$cg_{i,x} \rightarrow dg_z$" (or "$cg_{i,y} \rightarrow dg_z$") its general rule. We call decision rule "$cg_k \rightarrow dg_z$" meaningless if its confidence is less than or equal to the confidence in its general rule.

A granule mining oriented data warehousing model (Wan et al., 2007) was presented recently which facilitates the representations of multidimensional association rules. This model first constructs a decision table based on a set of selected attributes that meet the user constraints. It then builds up a multi-tier structure of granules based on the nature of these selected attributes. It also generates variant association mappings between granules to find association rules in order to satisfy what users want.

FUTURE TRENDS

Currently it is a big challenge to efficiently find interesting association rules in multidimensional databases for meeting what users want because the huge amount of frequent patterns and interesting association rules (Li and Zhong, 2006) (Wu et al., 2006) . Data mining oriented data warehousing techniques can be used to solve this challenging issue. OLAP technology dose not provide a mechanism to interpret the associations between data items, however, association mining can be conducted using OLAP via data cubes. Granule mining

provides an alternative way to represent association rules through multi-tier structures.

The key research issue is to allow users to semi-supervise the mining process and focus on a specified context from which rules can be extracted in multidimensional databases. It is possible to use data cubes and OLAP techniques to find frequent Meta-rules; however, it is an open issue to find only interesting and useful rules. Granule mining is a promising initiative for solving this issue. It must be interesting to combine granule mining and OLAP techniques for the development of efficient data mining oriented data warehousing models.

CONCLUSION

As mentioned above, it is a challenging task to significantly improve the quality of multidimensional association rule mining. The essential issue is how to represent meaningful multidimensional association rules efficiently. This chapter introduces the current achievements for solving this challenge. The basic idea is to represent discovered knowledge through using data warehousing techniques. Data mining oriented data warehousing can be simple classified into two areas: mining patterns and Meta rules through data cubes and mining granules and decision rules through multi-tier structures.

REFERENCES

Han, J. and Kamber, M. (2000), "*Data Mining: Concepts and Techniques*", Morgan Kaufmann Publishers, 2000.

Imielinski, T., Khachiyan, L., and Abdulghani, A. (2002), "Cubegrades: Generalizing association rules", *Data Mining and Knowledge Discovery*, 2002, **6(3):**219-258.

Inmon, W. H. (2005), "*Building the Data Warehouse*", Wiley Technology Publishing, 2005.

Kimball, R. and Ross, M. (2002), "*The Data Warehouse Toolkit - The Complete Guide to Dimensional Modelling*", 2nd edition, New York: John Wiley & Sons, 2002.

Li, Y. (2007), "Interpretations of Discovered Knowledge in Multidimensional Databases", in the Proceedings of 2007 *IEEE International Conference on Granular Computing*, Silicon Valley, US, pp. 307-312.

Li, Y., Yang, W. and Xu, Y. (2006), "Multi-tier granule mining for representations of multidimensional association rules", 6th *IEEE International Conference on Data Mining* (*ICDM*), Hong Kong, 2006, 953-958.

Li, Y. and Zhong, N. (2003), "Interpretations of association rules by granular computing", 3rd *IEEE International Conference on Data Mining* (*ICDM*), Florida, USA, 2003, 593-596

Li, Y. and Zhong, N. (2006), "Mining ontology for automatically acquiring Web user information needs", *IEEE Transactions on Knowledge and Data Engineering*, 2006, **18(4):** 554-568.

Messaoud, R. B., Rabaseda, S. L., Boussaid, O., and Missaoui, R. (2006), "Enhanced mining of association rules from data cubes", *The 9th ACM international workshop on Data warehousing and OLAP*, 2006, Arlington, Virginia, USA, pp. 11 – 18.

Pawlak, Z. (2002), "In pursuit of patterns in data reasoning from data, the rough set way," *3rd International Conference on Rough Sets and Current Trends in Computing*, USA, 2002, 1-9.

Pei, J., and Han, J. (2002), "Constrained frequent pattern mining: a pattern-growth view", *SIGKDD Exploration*, 2002, **4(1):** 31-39.

Wu, S.T., Li, Y. and Xu, Y. (2006), "Deploying Approaches for Pattern Refinement in Text Mining", 6th *IEEE International Conference on Data Mining* (ICDM 2006), Hong Kong, 2006, 1157-1161.

Xu, Y. and Li, Y. (2007), "Mining non-redundant association rule based on concise bases", *International Journal of Pattern Recognition and Artificial Intelligence*, 2007, 21(4): 659-675.

Yang, W., Li, Y., Wu, J. and Xu, Y. (2007), "Granule Mining Oriented Data Warehousing Model for Representations of Multidimensional Association Rules", accepted by *International Journal of Intelligent Information and Database Systems*, February 2007.

Zaki, M. J. (2004), "Mining non-redundant association rules," *Data Mining and Knowledge Discovery*, 2004, **9:** 223 -248.

KEY TERMS

Data Mining Oriented Data Warehousing: Refers to constructing such systems, in which both the data mining and data warehouse are a sub-component cooperating with each other. Using these systems, the data warehouse not only cleans and integrates data, but tailors data to fit the requirements of data mining. Thus, data mining becomes more efficient and accurate.

Decision Rule: A decision rule has the following format: "$X \rightarrow Y$", where X and Y are granules and ($X \wedge Y$) is a large granule, where small granule X is called condition granule and Y is called decision granule. Its support is the fraction of transactions that make both X and Y satisfy; and its confidence is the faction of transactions making X satisfy that also make Y satisfy.

Decision Table: A decision table is a tuple (T, V^T, C, D), where T is a set of objects (records) in which each record is a set of items, and $V^T = \{a_1, a_2, \ldots, a_n\}$ is a set of attributes, $C \cup D \subseteq V^T$ and $C \cap D = \varnothing$.

General Rule: Let "$X \rightarrow Y$" be a decision rule and $X = (X_1 \wedge X_2)$. We call "$X_1 \rightarrow Y$" (or "$X_2 \rightarrow Y$") a general rule of "$X \rightarrow Y$".

Granule: A granule is a predicate that describes common features of a set of objects (e.g., records, or transactions) for a selected set of attributes (or items).

Granule Mining: Granule mining is to find interesting associations between granules in databases.

Meaningless Rule: We call decision rule "$X \rightarrow Y$" meaningless if its confidence is less than or equals to the confidence in one of its general rules.

Meta-Rule: Ameta-rule in multidimensional databases usually has the following format: "$P_1 \wedge P_2 \wedge \ldots \wedge P_m \rightarrow Q_1 \wedge Q_2 \wedge \ldots \wedge Q_n$", where P_i and Q_j are some predicates which can be sets of data fields in different dimension levels.

Multi-Tier Structure: A multi-tier structure is a graph in which vertices are granules and edges are associations between granules. It used to organize discovered granules in multiple tires such that the granules in the same tier have the same size and in the different tiers have the different size.

Pattern Mining: That is the discovery of frequent patterns in databases.

Database Queries, Data Mining, and OLAP

Lutz Hamel
University of Rhode Island, USA

INTRODUCTION

Modern, commercially available relational database systems now routinely include a cadre of data retrieval and analysis tools. Here we shed some light on the interrelationships between the most common tools and components included in today's database systems: query language engines, data mining components, and on-line analytical processing (OLAP) tools. We do so by pair-wise juxtaposition which will underscore their differences and highlight their complementary value.

BACKGROUND

Today's commercially available relational database systems now routinely include tools such as SQL database query engines, data mining components, and OLAP (Craig, Vivona, & Bercovitch, 1999; Hamm, 2007; Melomed, Gorbach, Berger, & Bateman, 2006; Scalzo, 2003; Seidman, 2001). These tools allow developers to construct high powered business intelligence (BI) applications which are not only able to retrieve records efficiently but also support sophisticated analyses such as customer classification and market segmentation. However, with powerful tools so tightly integrated with the database technology understanding the differences between these tools and their comparative advantages and disadvantages becomes critical for effective application development. From the practitioner's point of view questions like the following often arise:

- Is running database queries against large tables considered data mining?
- Can data mining and OLAP be considered synonymous?
- Is OLAP simply a way to speed up certain SQL queries?

The issue is being complicated even further by the fact that data analysis tools are often implemented in terms of data retrieval functionality. Consider the data mining models in the Microsoft SQL server which are implemented through extensions to the SQL database query language (e.g. predict join) (Seidman, 2001) or the proposed SQL extensions to enable decision tree classifiers (Sattler & Dunemann, 2001). OLAP cube definition is routinely accomplished via the data definition language (DDL) facilities of SQL by specifying either a star or snowflake schema (Kimball, 1996).

MAIN THRUST OF THE CHAPTER

The following sections contain the pair wise comparisons between the tools and components considered in this chapter.

Database Queries vs. Data Mining

Virtually all modern, commercial database systems are based on the relational model formalized by Codd in the 60s and 70s (Codd, 1970) and the SQL language (Date, 2000) which allows the user to efficiently and effectively manipulate a database. In this model a database table is a representation of a mathematical relation, that is, a set of items that share certain characteristics or attributes. Here, each table column represents an attribute of the relation and each record in the table represents a member of this relation. In relational databases the tables are usually named after the kind of relation they represent. Figure 1 is an example of a table that represents the set or relation of all the customers of a particular store. In this case the store tracks the total amount of money spent by its customers.

Relational databases do not only allow for the creation of tables but also for the manipulation of the tables and the data within them. The most fundamental operation on a database is the query. This operation enables the user to retrieve data from database tables by asserting that the retrieved data needs to fulfill certain criteria. As an example, consider the fact that the store owner might be interested in finding out which customers spent more than $100 at the store. The fol-

Figure 1. A relational database table representing customers of a store

Id	*Name*	*ZIP*	*Sex*	*Age*	*Income*	*Children*	*Car*	*Total Spent*
5	Peter	05566	M	35	$40,000	2	Mini Van	$250.00
...	...	...	...	...	...	...	...	...
22	Maureen	04477	F	26	$55,000	0	Coupe	$50.00

lowing query returns all the customers from the above customer table that spent more than $100:

```
SELECT * FROM CUSTOMER_TABLE WHERE
TOTAL_SPENT > $100;
```

This query returns a list of all instances in the table where the value of the attribute *Total Spent* is larger than $100. As this example highlights, queries act as filters that allow the user to select instances from a table based on certain attribute values. It does not matter how large or small the database table is, a query will simply return all the instances from a table that satisfy the attribute value constraints given in the query. This straightforward approach to retrieving data from a database has also a drawback. Assume for a moment that our example store is a large store with tens of thousands of customers (perhaps an online store). Firing the above query against the customer table in the database will most likely produce a result set containing a very large number of customers and not much can be learned from this query except for the fact that a large number of customers spent more than $100 at the store. Our innate analytical capabilities are quickly overwhelmed by large volumes of data.

This is where differences between querying a database and mining a database surface. In contrast to a query which simply returns the data that fulfills certain constraints, data mining constructs models of the data in question. The models can be viewed as high level summaries of the underlying data and are in most cases more useful than the raw data, since in a business sense they usually represent understandable and actionable items (Berry & Linoff, 2004). Depending on the questions of interest, data mining models can take on very different forms. They include decision trees and decision rules for classification tasks, association rules for market basket analysis, as well as clustering for market segmentation among many other possible models. Good overviews of current data mining techniques and models can be found in (Berry & Linoff, 2004; Han & Kamber, 2001; Hand, Mannila, & Smyth, 2001; Hastie, Tibshirani, & Friedman, 2001).

To continue our store example, in contrast to a query, a data mining algorithm that constructs decision rules might return the following set of rules for customers that spent more than $100 from the store database:

```
IF AGE > 35 AND CAR = MINIVAN THEN
TOTAL SPENT > $100
```

or

```
IF SEX = M AND ZIP = 05566 THEN TOTAL
SPENT > $100
```

These rules are understandable because they summarize hundreds, possibly thousands, of records in the customer database and it would be difficult to glean this information off the query result. The rules are also actionable. Consider that the first rule tells the store owner that adults over the age of 35 that own a mini van are likely to spend more than $100. Having access to this information allows the store owner to adjust the inventory to cater to this segment of the population, assuming that this represents a desirable cross-section of the customer base. Similar with the second rule, male customers that reside in a certain ZIP code are likely to spend more than $100. Looking at census information for this particular ZIP code the store owner could again adjust the store inventory to also cater to this population segment presumably increasing the attractiveness of the store and thereby increasing sales.

As we have shown, the fundamental difference between database queries and data mining is the fact that in contrast to queries data mining does not return raw data that satisfies certain constraints, but returns models of the data in question. These models are attrac-

tive because in general they represent understandable and actionable items. Since no such modeling ever occurs in database queries we do not consider running queries against database tables as data mining, it does not matter how large the tables are.

Database Queries vs. OLAP

In a typical relational database queries are posed against a set of normalized database tables in order to retrieve instances that fulfill certain constraints on their attribute values (Date, 2000). The normalized tables are usually associated with each other via primary/foreign keys. For example, a normalized database of our store with multiple store locations or sales units might look something like the database given in Figure 2. Here, PK and FK indicate primary and foreign keys, respectively.

From a user perspective it might be interesting to ask some of the following questions:

- How much did sales unit A earn in January?
- How much did sales unit B earn in February?
- What was their combined sales amount for the first quarter?

Even though it is possible to extract this information with standard SQL queries from our database, the normalized nature of the database makes the formulation of the appropriate SQL queries very difficult. Furthermore, the query process is likely to be slow due to the fact that it must perform complex joins and multiple scans of entire database tables in order to compute the desired aggregates.

By rearranging the database tables in a slightly different manner and using a process called pre-aggregation or *computing cubes* the above questions can be answered with much less computational power enabling a real time analysis of aggregate attribute values – OLAP (Craig et al., 1999; Kimball, 1996; Scalzo, 2003). In order to enable OLAP, the database tables are usually arranged into a star schema where the inner-most table is called the fact table and the outer tables are called dimension tables. Figure 3 shows a star schema representation of our store organized along the main dimensions of the store business: customers, sales units, products, and time.

The dimension tables give rise to the dimensions in the pre-aggregated data cubes. The fact table relates the dimensions to each other and specifies the measures which are to be aggregated. Here the measures are "dollar_total", "sales_tax", and "shipping_charge". Figure 4 shows a three-dimensional data cube pre-aggregated from the star schema in Figure 3 (in this cube we ignored the customer dimension, since it is difficult to illustrate four-dimensional cubes). In the cube building process the measures are aggregated

Figure 2. Normalized database schema for a store

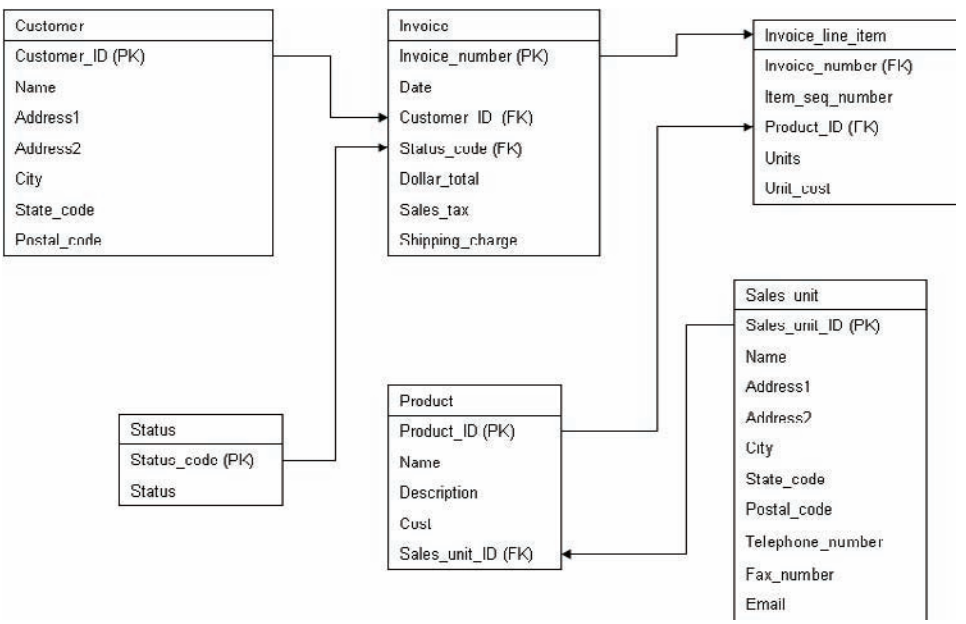


Figure 3. Star schema for a store database

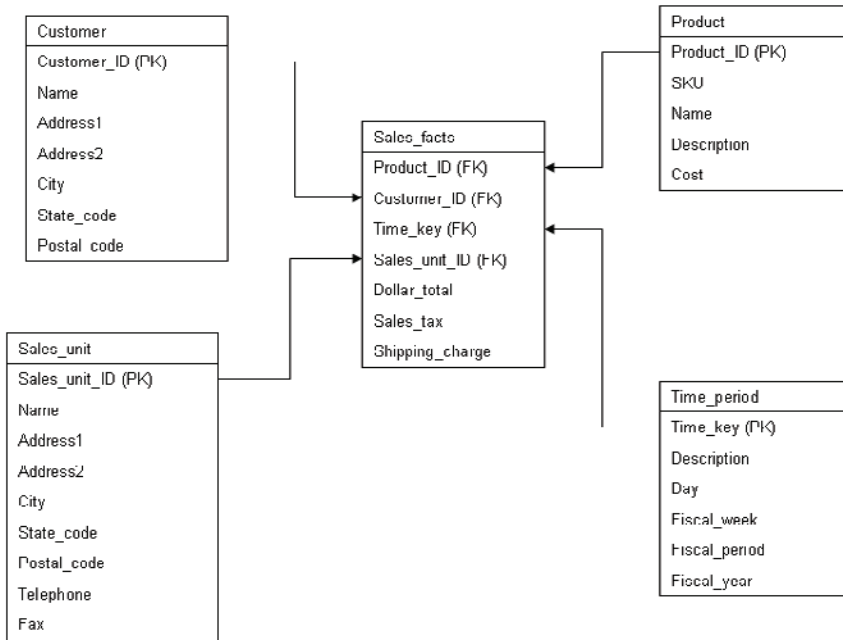
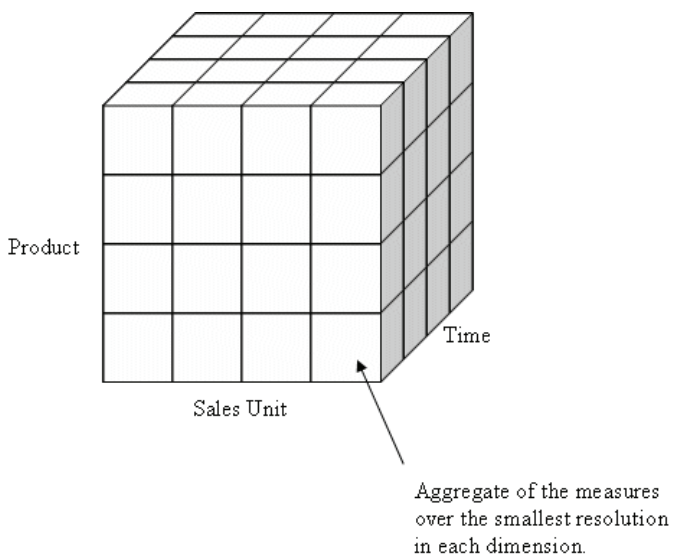


Figure 4. A three-dimensional data cube



along the smallest unit in each dimension giving rise to small pre-aggregated segments in a cube.

Data cubes can be seen as a compact representation of pre-computed query results[1]. Essentially, each segment in a data cube represents a pre-computed query result to a particular query within a given star schema. The efficiency of cube querying allows the user to interactively move from one segment in the cube to another enabling the inspection of query results in real time. Cube querying also allows the user to group and ungroup segments, as well as project segments onto given dimensions. This corresponds to such OLAP operations as roll-ups, drill-downs, and slice-and-dice, respectively (Gray, Bosworth, Layman, & Pirahesh, 1997). These specialized operations in turn provide answers to the kind of questions mentioned above.

As we have seen, OLAP is enabled by organizing a relational database in a way that allows for the pre-aggregation of certain query results. The resulting data cubes hold the pre-aggregated results giving the user the ability to analyze these aggregated results in real time using specialized OLAP operations. In a larger context we can view OLAP as a methodology for the organization of databases along the dimensions of a business making the database more comprehensible to the end user.

Data Mining vs. OLAP

Is OLAP data mining? As we have seen, OLAP is enabled by a change to the data definition of a relational database in such a way that it allows for the pre-computation of certain query results. OLAP itself is a way to look at these pre-aggregated query results in real time. However, OLAP itself is still simply a way to evaluate queries which is different from building models of the data as in data mining. Therefore, from a technical point of view we cannot consider OLAP to be data mining. Where data mining tools model data and return actionable rules, OLAP allows users to compare and contrast measures along business dimensions in real time.

It is interesting to note, that recently a tight integration of data mining and OLAP has occurred. For example, Microsoft SQL Server 2000 not only allows OLAP tools to access the data cubes but also enables its data mining tools to mine data cubes (Seidman, 2001).

FUTURE TRENDS

Perhaps the most important trend in the area of data mining and relational databases is the liberation of data mining tools from the "single table requirement." This new breed of data mining algorithms is able to take advantage of the full relational structure of a relational database obviating the need of constructing a single table that contains all the information to be used in the data mining task (Dézeroski & Lavraéc, 2001; Getoor, L., & Taskar, B., 2007). This allows for data mining tasks to be represented naturally in terms of the actual database structures, e.g. (Yin, Han, Yang, & Yu, 2004), and also allows for a natural and tight integration of data mining tools with relational databases.

CONCLUSION

Modern, commercially available relational database systems now routinely include a cadre of data retrieval and analysis tools. Here, we briefly described and contrasted the most often bundled tools: SQL database query engines, data mining components, and OLAP tools. Contrary to many assertions in the literature and business press, performing queries on large tables or manipulating query data via OLAP tools is not considered data mining due to the fact that no data modeling occurs in these tools. On the other hand, these three tools complement each other and allow developers to pick the tool that is right for their application: queries allow ad hoc access to virtually any instance in a database; data mining tools can generate high-level, actionable summaries of data residing in database tables; and OLAP allows for real-time access to pre-aggregated measures along important business dimensions. In this light it does not seem surprising that all three tools are now routinely bundled.

NOTE

Figures 2 and 3 are based on Figures 3.2 and 3.3 from (Craig et al., 1999), respectively.

REFERENCES

Berry, M. J. A., & Linoff, G. S. (2004). *Data mining techniques : For marketing, sales, and customer relationship management* (2nd ed.): John Wiley & Sons.

Codd, E. F. (1970). A relational model of data for large shared data banks. *Communications of the ACM, 13*(6), 377-387.

Craig, R. S., Vivona, J. A., & Bercovitch, D. (1999). *Microsoft data warehousing*: John Wiley & Sons.

Date, C. J. (2000). *An introduction to database systems* (7th ed.). Reading, MA: Addison-Wesley.

Dézeroski, S., & Lavraéc, N. (2001). *Relational data mining*. New York: Springer.

Getoor, L., & Taskar, B., (Eds.) (2007). *Introduction to statistical relational learning*, MIT Press.

Gray, J., Bosworth, A., Layman, A., & Pirahesh, H. (1997). Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals. *Data Mining and Knowledge Discovery, 1*(1), 29-53.

Hamm, C. (2007). *Oracle data mining*. Rampant Press.

Han, J., & Kamber, M. (2001). *Data mining: Concepts and techniques*. San Francisco: Morgan Kaufmann Publishers.

Hand, D. J., Mannila, H., & Smyth, P. (2001). *Principles of data mining*. Cambridge, MA: MIT Press.

Hastie, T., Tibshirani, R., & Friedman, J. H. (2001). *The elements of statistical learning: Data mining, inference, and prediction*. New York: Springer.

Kimball, R. (1996). *The data warehouse toolkit: Practical techniques for building dimensional data warehouses*. New York: John Wiley & Sons.

Melomed, E., Gorbach, I., Berger, A., & Bateman, P. (2006). *Microsoft SQL Server 2005 Analysis Services*. Sams.

Pendse, N. (2001). *Multidimensional data structures*, from http://www.olapreport.com/MDStructures.htm

Sattler, K., & Dunemann, O. (2001, November 5-10). *SQL database primitives for decision tree classifiers.* Paper presented at the 10th International Conference on Information and Knowledge Management, Atlanta, Georgia.

Scalzo, B. (2003). *Oracle DBA guide to data warehousing and star schemas*. Upper Saddle River, N.J.: Prentice Hall PTR.

Seidman, C. (2001). *Data Mining with Microsoft SQL Server 2000 Technical Reference*. Microsoft Press.

Yin, X., Han, J., Yang, J., & Yu, P. S. (2004). *CrossMine: Efficient classification across multiple database relations.* Paper presented at the 20th International Conference on Data Engineering (ICDE 2004), Boston, MA.

KEY TERMS

D

Business Intelligence: Business intelligence (BI) is a broad category of technologies that allows for gathering, storing, accessing and analyzing data to help business users make better decisions. (Source: http://www.oranz.co.uk/glossary_text.htm)

Data Cubes: Also known as OLAP cubes. Data stored in a format that allows users to perform fast multi-dimensional analysis across different points of view. The data is often sourced from a data warehouse and relates to a particular business function. (Source: http://www.oranz.co.uk/glossary_text.htm)

OLAP: On-Line Analytical Processing - a category of applications and technologies for collecting, managing, processing and presenting multidimensional data for analysis and management purposes. (Source: http://www.olapreport.com/glossary.htm)

Normalized Database: A database design that arranges data in such a way that it is held at its lowest level avoiding redundant attributes, keys, and relationships. (Source: http://www.oranz.co.uk/glossary_text.htm)

Query: This term generally refers to databases. A query is used to retrieve database records that match certain criteria. (Source: http://usa.visa.com/business/merchants/online_trans_glossary.html)

SQL: Structured Query Language - SQL is a standardized programming language for defining, retrieving, and inserting data objects in relational databases.

Star Schema: A database design that is based on a central detail fact table linked to surrounding dimension tables. Star schemas allow access to data using business terms and perspectives. (Source: http://www.ds.uillinois.edu/glossary.asp)

ENDNOTE

[1] Another interpretation of data cubes is as an effective representation of multidimensional data along the main business dimensions (Pendse, 2001).

Database Sampling for Data Mining

Patricia E.N. Lutu
University of Pretoria, South Africa

INTRODUCTION

In data mining, sampling may be used as a technique for reducing the amount of data presented to a data mining algorithm. Other strategies for data reduction include dimension reduction, data compression, and discretisation. For sampling, the aim is to draw, from a database, a random sample, which has the same characteristics as the original database. This chapter looks at the sampling methods that are traditionally available from the area of statistics, how these methods have been adapted to database sampling in general, and database sampling for data mining in particular.

BACKGROUND

Given the rate at which database / data warehouse sizes are growing, attempts at creating faster / more efficient algorithms that can process massive data sets, may eventually become futile exercises. Modern database and data warehouse sizes are in the region of 10's or 100's of terabytes, and sizes continue to grow. A query issued on such a database / data warehouse could easily return several millions of records.

While the costs of data storage continue to decrease, the analysis of data continues to be hard. This is the case for even traditionally simple problems requiring aggregation, for example, the computation of a mean value for some database attribute. In the case of data mining, the computation of very sophisticated functions, on very large number of database records, can take several hours, or even days. For inductive algorithms, the problem of lengthy computations is compounded by the fact that many iterations are needed in order to measure the training accuracy as well as the generalization accuracy.

There is plenty of evidence to suggest that for inductive data mining, the learning curve flattens after only a small percentage of the available data, from a large data set has been processed (Catlett 1991; Kohavi 1996; Provost et al, 1999). The problem of overfitting (Dietterich, 1995), also dictates that the mining of massive data sets should be avoided. The sampling of databases has been studied by researchers for some time. For data mining, sampling should be used as a data reduction technique, allowing a data set to be represented by a much smaller random sample that can be processed much more efficiently.

MAIN THRUST OF THE CHAPTER

There are a number of key issues to be considered before obtaining a suitable random sample for a data mining task. It is essential to understand the strengths and weaknesses of each sampling method. It is also essential to understand which sampling methods are more suitable to the type of data to be processed and the data mining algorithm to be employed. For research purposes, we need to look at a variety of sampling methods used by statisticians, and attempt to adapt them to sampling for data mining.

Some Basics of Statistical Sampling Theory

In statistics, the theory of sampling, also known as statistical estimation, or the representative method, deals with the study of suitable methods of selecting a representative sample of a population, in order to study or estimate values of specific characteristics of the population (Neyman, 1934). Since the characteristics being studied can only be estimated from the sample, confidence intervals are calculated to give the range of values within which the actual value will fall, with a given probability.

There are a number of sampling methods discussed in the literature, for example the book by Rao (Rao, 2000). Some methods appear to be more suited to

database sampling that others. Simple random sampling (SRS) stratified random sampling and cluster sampling are three such methods. Simple random sampling involves selecting at random, elements of the population, P, to be studied. The method of selection may be either with replacement (SRSWR) or without replacement (SRSWOR). For very large populations, however, SRSWR and SRSWOR are equivalent. For simple random sampling, the probabilities of inclusion of the elements may or may not be uniform. If the probabilities are not uniform then a weighted random sample is obtained.

The second method of sampling is stratified random sampling. Here, before the samples are drawn, the population P, is divided into several strata, p_1, p_2,.. p_k, and the sample S is composed of k partial samples s_1, s_2,.., s_k, each drawn randomly, with replacement or not, from one of the strata. Rao (2000) discusses several methods of allocating the number of sampled elements for each stratum. Bryant et al (1960) argue that, if the sample is allocated to the strata in proportion to the number of elements in the strata, it is virtually certain that the stratified sample estimate will have a smaller variance than a simple random sample of the same size. The stratification of a sample may be done according to one criterion. Most commonly though, there are several alternative criteria that may be used for stratification. When this is the case, the different criteria may all be employed to achieve multi-way stratification. Neyman (1934) argues that there are situations when it is very difficult to use an individual unit as the unit of sampling. For such situations, the sampling unit should be a group of elements, and each stratum should be composed of several groups. In comparison with stratified random sampling, where samples are selected from each stratum, in cluster sampling, a sample of clusters is selected and observations/measurements are made on the clusters. Cluster sampling and stratification may be combined (Rao, 2000).

Database Sampling Methods

Database sampling has been practiced for many years for purposes of estimating aggregate query results, database auditing, query optimization, and, obtaining samples for further statistical processing (Olken, 1993). Static sampling (Olken, 1993) and adaptive (dynamic) sampling (Haas & Swami, 1992) are two alternatives for obtaining samples for data mining tasks. In recent years, many studies have been conducted in applying sampling to inductive and non-inductive data mining (John & Langley, 1996; Provost et al, 1999; Toivonen, 1996).

Simple Random Sampling

Simple random sampling is by far, the simplest method of sampling a database. Simple random sampling may be implemented using sequential random sampling or reservoir sampling. For sequential random sampling, the problem is to draw a random sample of size n without replacement, from a file containing N records. The simplest sequential random sampling method is due to Fan et al (1962) and Jones (1962). An independent uniform random variate (from the uniform interval (0,1)) is generated for each record in the file to determine whether the record should be included in the sample. If m records have already been chosen from among the first t records in the file, the $(t+1)^{st}$ record is chosen with probability (RQsize / RMsize), where RQsize = (n-m) is the number of records that still need to be chosen for the sample, and RMsize = (N-t) is the number of records in the file, still to be processed. This sampling method is commonly referred to as method S (Vitter, 1987).

The *reservoir sampling* method (Fan et al, 1962; Jones, 1962; Vitter, 1985 & 1987) is a sequential sampling method over a finite population of database records, with an unknown population size. Olken, 1993) discuss its use in sampling of database query outputs on the fly. This technique produces a sample of size S, by initially placing the first S records of the database/file/query in the reservoir. For each subsequent k^{th} database record, that record is accepted with probability S / k. If accepted, it replaces a randomly selected record in the reservoir.

Acceptance/Rejection sampling (A/R sampling) can be used to obtain *weighted samples* (Olken 1993). For a weighted random sample, the probabilities of inclusion of the elements of the population are not uniform. For database sampling, the inclusion probability of a data record is proportional to some weight calculated from the record's attributes. Suppose that one database record r_j is to be drawn from a file of n records with the probability of inclusion being proportional to the weight w_j. This may be done by generating a uniformly distributed random integer j, $1 \leq j \leq n$ and then accepting the sampled record r_j with probability $\alpha_j = w_j /$

w_{max}, where w_{max} is the maximum possible value for w_j. The acceptance test is performed by generating another uniform random variate u_j, $0 \leq u_j \leq 1$, and accepting r_j iff $u_j < \alpha_j$. If r_j is rejected, the process is repeated until some r_j is accepted.

Stratified Sampling

Density biased sampling (Palmer & Faloutsos, 2000), is a method that combines clustering and stratified sampling. In density biased sampling, the aim is to sample so that within each cluster points are selected uniformly, the sample is density preserving, and sample is biased by cluster size. Density preserving in this context means that the expected sum of weights of the sampled points for each cluster is proportional to the cluster's size. Since it would be infeasible to determine the clusters apriori, groups are used instead, to represent all the regions in n-dimensional space. Sampling is then done to be density preserving for each group. The groups are formed by 'placing' a d-dimensional grid over the data. In the d-dimensional grid, the d dimensions of each cell are labeled either with a bin value for numeric attributes, or by a discrete value for categorical attributes. The d-dimensional grid defines the strata for multi-way stratified sampling. A one-pass algorithm is used to perform the weighted sampling, based on the reservoir algorithm.

Adaptive Sampling

Lipton et al (1990) use adaptive sampling, also known as sequential sampling, for database sampling. In sequential sampling, a decision is made after each sampled element, whether to continue sampling. Olken (1993) has observed that sequential sampling algorithms outperform conventional single-stage algorithms , in terms of the number of sample elements required, since they can adjust the sample size to the population parameters. Haas and Swami (1992) have proved that sequential sampling uses the minimum sample size for the required accuracy.

John and Langley (1996) have proposed a method they call dynamic sampling, which combines database sampling with the estimation of classifier accuracy. The method is most efficiently applied to classification algorithms which are incremental, for example naïve Bayes and artificial neural-network algorithms such as backpropagation. They define the concept of 'probably close enough' (PCE), which they use for determining when a sample size provides an accuracy that is probably good enough. 'Good enough' in this context means that there is a small probability δ that the mining algorithm could do better by using the entire database. The smallest sample size n, is chosen from a database of size N, so that: $Pr(acc(N) - acc(n) > \varepsilon) <= \delta$, where: acc(n) is the accuracy after processing a sample of size n, and ε is a parameter that describes what 'close enough' means. The method works by gradually increasing the sample size n until the PCE condition is satisfied.

Provost, Jensen and Oates (1999) use progressive sampling, another form of adaptive sampling, and analyse its efficiency relative to induction with all available examples. The purpose of progressive sampling is to establish n_{min}, the size of the smallest sufficient sample. They address the issue of convergence, where convergence means that a learning algorithm has reached its plateau of accuracy. In order to detect convergence, they define the notion of a sampling schedule S as $S = \{n_o, n_1, .., n_k\}$ where n_i is an integer that specifies the size of the sample, and S is a sequence of sample sizes to be provided to an inductive algorithm. They show that schedules which n_i increases geometrically as: $\{no, a.n_o, a^2.n_o, .., a^k.n_o\}$, are asymptotically optimal. As one can see, progressive sampling is similar the adaptive sampling method of John and Langley (1996), except that a non-linear increment for the sample size is used.

THE SAMPLING PROCESS

Several decisions need to be made when sampling a database. One needs to decide on a sampling method, a suitable sample size, the a level accuracy that can be tolerated. These issues are discussed below.

Deciding on a Sampling Method

The data to be sampled may be balanced, imbalanced, clustered or unclustered. These characteristics will affect the quality of the sample obtained. While simple random sampling is very easy to implement, it may produce non-representative samples for data that is imbalanced or clustered. On the other hand, stratified sampling, with a good choice of strata cells, can be used to produce representative samples, regardless

of the characteristics of the data. Implementation of one-way stratification should be straight forward, however, for multi-way stratification, there are many considerations to be made. For example, in density-biased sampling, a d-dimensional grid is used. Suppose each dimension has n possible values (or bins). The multi-way stratification will result in n^d strata cells. For large d, this is a very large number of cells. When it is not easy to estimate the sample size in advance, adaptive (or dynamic) sampling may be employed, if the data mining algorithm is incremental.

Determining the Representative Sample Size

For static sampling, the question must be asked: 'What is the size of a representative sample?'. A sample is considered statistically valid if it is sufficiently similar to the database from where it is drawn (John & Langley, 1996). Univariate sampling may be used to test that each field in the sample comes from the same distribution as the parent database. For categorical fields, the chi-squared test can be used to test the hypothesis that the sample and the database come from the same distribution. For continuous-valued fields, a 'large-sample' test can be used to test the hypothesis that the sample and the database have the same mean. It must however be pointed out that, obtaining fixed size representative samples from a database is not a trivial task, and, consultation with a statistician is recommended.

For inductive algorithms, the results from the theory of probably approximately correct (PAC) learning have been suggested in the literature (Valiant, 1984; Haussler, 1990). These have however been largely criticized for overestimating the sample size (eg. Haussler, 1990). For incremental inductive algorithms, dynamic sampling (John & Langley, 1996; Provost et al, 1999) may be employed to determine when a sufficient sample has been processed. For association rule mining, the methods described by Toivonen (1996) may be used to determine the sample size.

Determining the Accuracy and Confidence of Results Obtained from Samples

For the task of inductive data mining, suppose we have estimated the classification error for a classifier constructed from sample S, to be $error_s$ (h), as the proportion of the test examples that are misclassified. Statistical theory, based on the central limit theorem, enables us to conclude that, with approximately N% probability, the true error lies in the interval:

$$Error_s(h) \pm Z_N \sqrt{(Error_s(h)(1-Error_s(h))/n)}$$

As an example, for a 95% probability, the value of Z_N is 1.96. Note that, $Error_S(h)$ is the mean error and $(Error_S(h)(1-Error_S(h))/n)$ is the variance of the error. Mitchell (1997, chapter 5) gives a detailed discussion of the estimation of classifier accuracy and confidence intervals. For non–inductive data mining algorithms, there are known ways of estimating the error and confidence in the results obtained from samples. For association rule mining, for example Toivonen (1996) states that the lower bound for the size of the sample, given an error bound ε and a maximum probability δ is given by:

$$|s| \geq \frac{1}{2\varepsilon^2} \ln \frac{2}{\delta}$$

The value ε is the error in estimating the frequencies of frequent item sets for some give set of attributes.

FUTURE TRENDS

There two main thrusts in research on establishing the sufficient sample size for a data mining task. The theoretical approach, most notably the PAC framework and the Vapnik-Chernonenkis (VC) dimension have produced results which are largely criticized by data mining researchers and practitioners. However, research will continue in this area, and may eventually yield practical guidelines. On the empirical front, researchers will continue to conduct simulations that will lead to generalizations on how to establish sufficient sample sizes.

CONCLUSIONS

Modern databases and data warehouses are massive, and will continue to grow in size. It is essential for researchers to investigate data reduction techniques that can greatly reduce the amount of data presented

to a data mining algorithm. More than fifty years of research has produced a variety of sampling algorithms for database tables and query result sets. The selection of a sampling method should depend on the nature of the data as well as the algorithm to be employed. Estimation of sample sizes for static sampling, is a tricky issue. More research in this area is needed in order to provide practical guidelines. Stratified sampling would appear to be a versatile approach to sampling any type of data. However, more research is needed to address especially the issue of how to define the strata for sampling.

REFERENCES

Bryant, E.C., Hartley, H.O. & Jessen, R.J. (1960). Design and estimation in two-way stratification. *Journal of the American Statistical Association, 55,* 105-124.

Catlett, J. (1991). Megainduction: a test flight. Proc. Eighth Workshop on Machine Learning. Morgan Kaufman, 596-599.

Dietterich, T. (1995). Overfitting and undercomputing in machine learning. *ACM Computing Surveys, 27*(3), 326-327.

Fan, C., Muller, M , & Rezucha, I. (1962). Development of sampling plans by using

sequential (item by item) selection techniques and digital computers. *Journal of the American Statistical Association, 57,* 387-402.

Haas, P.J. & Swami, A.N. (1992). *Sequential sampling procedures for query size estimation*. IBM Technical Report RJ 8558, IBM Alamaden.

Haussler, D., (1990). *Probably approximately correct learning*. National Conference on Artificial Intelligence.

John, G.H, & Langley, P. (1996). Static versus dynamic sampling for data mining. *Proc. Second International Conference on Knowledge Discovery in Databases and Data Mining*. AAAI/MIT Press.

Jones, T. (1962). A note on sampling from tape files. *Communications of the ACM, 5*, 343-343.

Kohavi, R. (1996). Scaling up the accuracy on naïve-Bayes classifiers: a decision tree hybrid. *Proc. Second International Conference on Knowledge Discovery and Data Mining.*

Lipton, R., Naughton, J., & Schneider, D., (1990). Practical selectivity estimation through adaptive sampling. *ACM SIGMOD International Conference on the Management of Data,* 1-11.

Mitchell T.M. (1997). *Machine Learning*. McGraw-Hill.

Neyman, J. (1934). On the two different aspects of the representative method: the method of stratified sampling and the method of purposive selection. *Journal of the Royal Statistical Society, 97*, 558-625.

Olken F. (1993). *Random Sampling from Databases.* PhD thesis, University of California at Berkeley.

Palmer C.R. & Faloutsos, C. (2000). Density biased sampling: An improved method for data mining and clustering', *Proceedings of the ACM SIGMOD Conference, 2000,* 82-92.

Provost, F., Jensen, D., & Oates, T. (1999). Efficient progressive sampling. *Proc. Fifth ACM DIGKDD International Conference on Knowledge Discovery and Data Mining,* San Diego, CA, 23-32.

Rao, P.S.R.S. (2000). *Sampling Methodologies with Applications.* Chapman & Hall/CRC, Florida.

Toivonen, H. (1996). Sampling large databases for association rules. *Proc. Twenty-second Conference on Very Large Databases* – VLDDB96, Mumbai India.

Valiant, L.G. (1984). A theory of the learnable. *Communications of the ACM, 27*(11), 1134-1142.

Vitter, J.S. (1985). Random sampling with a reservoir. *ACM Transactions on Mathematical Software, 11*, 37-57.

Vitter, J.S. (1987). An efficient method for sequential random sampling. *ACM Transactions on Mathematical Software, 13*(1), 58-67.

KEY TERMS

Cluster Sampling: In cluster sampling, a sample of clusters is selected and observations / measurements are made on the clusters.

Density-Biased Sampling: A database sampling method that combines clustering and stratified sampling.

Dynamic Sampling (Adaptive Sampling): A method of sampling where sampling and processing of data proceed in tandem. After processing each incremental part f the sample, a decision is made whether to continue sampling or not.

Reservoir Sampling: A database sampling method that implements uniform random sampling on a database table of unknown size, or a query result set of unknown size.

Sequential Random Sampling: A database sampling method that implements uniform random sampling on a database table whose size is known.

Simple Random Sampling: Simple random sampling involves selecting at random, elements of the population to be studied. The sample S, is obtained by selecting at random, single elements of the population P.

Simple Random Sampling With Replacement (SRSWR): A method of simple random sampling where an element stands a chance of being selected more than once.

Simple Random Sampling Without Replacement (SRSWOR): A method of simple random sampling where each element stands a chance of being selected only once.

Static Sampling: A method of sampling where the whole sample is obtained before processing begins. The user must specify the sample size.

Stratified Sampling: For this method, before the samples are drawn, the population P, is divided into several strata, $p_1, p_2,.. p_k$, and the sample S is composed of k partial samples $s_1, s_2,.., s_k$, each drawn randomly, with replacement or not, from one of the strata.

Uniform Random Sampling: A method of simple random sampling where the probabilities of inclusion for each element are equal.

Weighted Random Sampling: A method of simple random sampling where the probabilities of inclusion for each element are not equal.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 344-348, copyright 2005 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

Database Security and Statistical Database Security

Edgar R. Weippl
Secure Business Austria, Austria

INTRODUCTION

In this article we will present an introduction to issues relevant to database security and statistical database security. We will briefly cover various security models, elaborate on how data analysis in data warehouses (DWH) might compromise an individual's privacy, and explain which safeguards can be used to prevent attacks.

In most companies, databases are an essential part of IT infrastructure since they store critical business data. In the last two decades, databases have been used to process increasing amounts of transactional data, such as, a complete account of a person's purchases from a retailer or connection data from calls made on a cell phone.

As soon as this data became available from transactional databases and online transactional processing (OLTP) became well established, the next logical step was to use the knowledge contained in the vast amounts of data. Today, data warehouses (DWH) store aggregated data in an optimal way to serve queries related to business analysis.

In recent years, most people have begun to focus their attention on security. Early OLTP applications were mainly concerned with integrity of data during transactions; today privacy and secrecy are more important as databases store an increasing amount of information about individuals, and data from different systems can be aggregated. Thuraisingham (2002) summarizes the requirements briefly as "*However, we do not want the information to be used in an incorrect manner."*

All security requirements stem from one of three basic requirements: confidentiality (aka secrecy), integrity, and availability (CIA). Confidentiality refers to the requirement that only authorized subjects, that is, people or processes should be permitted to read data. Integrity means that unauthorized modifications must not be permitted. This includes both modifications by unauthorized people and incorrect modification by authorized users. To correctly perform the services requested, the system needs to remain available; a denial-of-service compromises the requirement of availability.

Other security requirements may include privacy, non-repudiation, and separation of duties. These requirements are, however, composite requirements that can be traced back to one of the three basic requirements. Privacy, for instance, is the non-disclosure (=confidentiality) of personal data; non-repudiation refers to the integrity of transaction logs and integrity of origin. Throughout this article we will focus only on technical attacks and safeguards and not on social engineering. Social engineering is often the easiest and, in many cases, a very successful attack vector. For an in-depth coverage of social engineering we recommend (Böck, 2007).

In Section 2 we cover the most relevant access control models; in Section 3 we provide an overview of security in statistical databases. Finally, in Section 4 we highlight the essentials of securing not only the transactional and the statistical databases but the entire system.

BACKGROUND

Access Control is the most important technique or mechanism for implementing the requirements of confidentiality and integrity. Since databases were among the first large-scale systems in military applications, there is a long history of security models, dating back to the 1960s. The basic principle in all access control models is that a *subject* is or is not permitted to perform a certain *operation* on an *object*. This process is described by the triplet (s, op, o). A security policy specifies who is authorized to do what. A security mechanism allows enforcement of a chosen security policy.

One can distinguish between two fundamentally different access control mechanisms: discretionary access control (DAC) and mandatory access control (MAC). In DAC models the user decides which subject is able

to access which object to perform a certain operation. In contrast, when using MAC, the system decides who is allowed to access which resource and the individual user has no discretion to decide on access rights.

Discretionary Access Control (DAC)

In relational database management systems (DBMS), the objects that need to be protected are tables and views. Modern DBMS allow a fine granularity of access control so that access to individual fields of a record can be controlled.

By default, a subject has no access. Subjects may then be *granted* access, which can be *revoked* anytime. In most systems the creator of a table or a view is automatically granted all privileges related to it. The DBMS keeps track of who subsequently gains and loses privileges, and ensures that only requests from subjects who have the required privileges—at the time the request is executed—are allowed.

Mandatory Access Control (MAC)

Mandatory Access Control is based on system-wide policies that cannot be changed by individual users. Each object in the database is automatically assigned a security class based on the access privileges of the user who created the object.

The most widely known implementation of a MAC system is a multi-level security (MLS) system. MLS systems were first described by Bell LaPadula (Bell, 1975) in the 1960s. Each subject, which could either be a user or user program, is assigned a *clearance* for a security class. Objects are assigned security *levels*. Security levels and clearances can be freely defined as long as all items are comparable pair-wise. Most common are security classes (i.e., levels and clearances), such as, top secret (TS), secret (S), confidential (C), and unclassified (U).

Rules based on security levels and clearances govern who can read or write which objects. Today, there are only a few commercially available systems that support MAC, such as, SELinux or also Oracle DBMS (Version 9 and higher) when the Oracle Label Security (OLS) option is installed.

The main reason to use a MAC system is that it prevents inherent flaws of discretionary access control, which are commonly referred to as the Trojan horse problem. The user Alice creates a program and gives Bob INSERT privileges for the table mySecret. Bob knows nothing about this. Alice modifies the code of an executable that Bob uses so that it additionally writes Bob's secret data to the table mySecret. Now, Alice can see Bob's secret data. While the modification of the application code is beyond the DBMS' control, it can still prevent the use of the database as a channel for secret information.

ACCESS CONTROL FOR RELATIONAL DATABASES

Role-Based Access Control (RBAC)

With RBAC (Sandhu, 2000), system administrators create roles according to the job functions defined in a company; they grant permissions to those roles and subsequently assign users to the roles on the basis of their specific job responsibilities and qualifications. Thus, roles define the authority of users, the competence that users have, and the trust that the company gives to the user. Roles define both, the specific individuals allowed to access objects and the extent to which or the mode in which they are allowed to access the object (see Sandhu & Coyne & Feinstein & Youman, 1996). Access decisions are based on the roles a user has activated (Sandhu & Ferraiolo & Kuhn, 2000).

The basic RBAC model consists of four entities: users, roles, permissions, and sessions. A user is a subject who accesses different, protected objects. A role is a named job function that describes the authority, trust, responsibility, and competence of a role member. A permission is an approval for a particular type of access to one or more objects. Permissions describe which actions can be performed on a protected object and may apply to one or more objects. Both permissions and users are assigned to roles. These assignments, in turn, define the scope of access rights a user has with respect to an object. By definition, the user assignment and permission assignment relations are many-to-many relationships.

Users establish sessions during which they may activate a subset of the roles they belong to. A session maps one user to many possible roles, which results in the fact that multiple roles can be activated simultaneously and every session is assigned with a single user. Moreover, a user might have multiple sessions opened simultaneously. Belonging to several roles, a user can

invoke any subset of roles to accomplish a given task. In other words, sessions enable a dynamic activation of user privileges (see Sandhu & Coyne & Feinstein & Youman, 1996).

We will briefly summarize various properties of the NISTs RBAC model as pointed out by Sandhu et al. (Sandhu & Ferraiolo & Kuhn, 2000). RBAC does not define the degree of scalability implemented in a system with respect to the number of roles, number of permissions, size of role hierarchy, or limits on user-role assignments, etc.

Coexistence with MAC / DAC

Mandatory access control (MAC) is based on distinct levels of security to which subjects and objects are assigned. Discretionary access control (DAC) controls access to an object on the basis of an individual user's permissions and/or prohibitions. RBAC, however, is an independent component of these access controls and can coexist with MAC and DAC. RBAC can be used to enforce MAC and DAC policies, as shown in (Osborn & Sandhu & Munawer, 2000). The authors point out the possibilities and configurations necessary to use RBAC in the sense of MAC or DAC.

Levels Defined in the NIST Model of RBAC

The NIST Model of RBAC is organized into four levels of increasing functional capabilities as mentioned above: (1) flat RBAC, (2) hierarchical RBAC, (3) constrained RBAC, and (4) symmetric RBAC. These levels are cumulative such that each adds exactly one new requirement. The following subsections will offer a brief presentation of the four levels.

The basic principle is that users are assigned to roles (user-role assignment, indicated through the membership association), permissions are assigned to roles (permission-role assignment, indicated through the authorization association) and users gain permissions defined in the role(s) they activate. A user can activate several roles within a session (indicated through the n-ary activation association). As all these assignments are many-to-many relationships; a user can be assigned to many roles and a role can contain many permissions. Flat RBAC requires a user-role review whereby the roles assigned to a specific user can be determined as well as users assigned to a specific role. Similarly, flat RBAC requires a permission-role review. Finally, flat RBAC requires that users can simultaneously use permissions granted to them via multiple roles.

Flat RBAC represents the traditional group-based access control as it can be found in various operating systems (e.g., Novell Netware, Windows NT). The requirements of flat RBAC are obvious and obligatory for any form of RBAC. According to (Sandhu & Ferraiolo & Kuhn, 2000), the main issue in defining flat RBAC is to determine which features to exclude.

Hierarchical RBAC supports role hierarchies built using the sub-role and super-role association. A hierarchy defines a seniority relation between roles, whereby senior roles acquire the permissions of their juniors.

Role hierarchies may serve three different purposes:

- *Inheritance hierarchies* whereby activation of a role implies activation of all junior roles.
- *Activation hierarchies* whereby there is no implication of the activation of all junior roles (each junior role must be explicitly activated to enable its permissions in a session); or a combination of both.

Constrained RBAC supports separation of duties (SOD). SOD is a technique for reducing the possibility of fraud and accidental damage. It spreads responsibility and authority for an action or task over multiple users thereby reducing the risk of fraud.

Symmetric RBAC adds requirements for permission-role review similar to the user-role review introduced in flat RBAC. Thus, the roles to which a particular permission is assigned can be determined as well as permissions assigned to a specific role. However, implementing this requirement in large-scale distributed systems may be a very complex challenge.

Usage Control

According to Bertino et al. (2006), secure data mining is an aspect of secure knowledge management. The authors emphasize the importance of usage control (cf. Park 2004) to implement continuous decisions of whether a resource may or may not be used. Similar to the reference monitor in classical access control, Thuraisingham (2005) proposed to use a privacy controller to "*limit and watch access to the DBMS (that access the data in the database)*." She sees a privacy constraint as a form of integrity constraint, and proposes to use mechanisms

of the DBMS for guaranteeing integrity: "*In these techniques, some integrity constraints, which are called derivation rules, are handled during query processing, some integrity constraints, known as integrity rules, are handled during database updates, and some integrity constraints, known as schema rules, are handled during database design.*" Thuraisingham (2005)

STATISTICAL DATABASE SECURITY

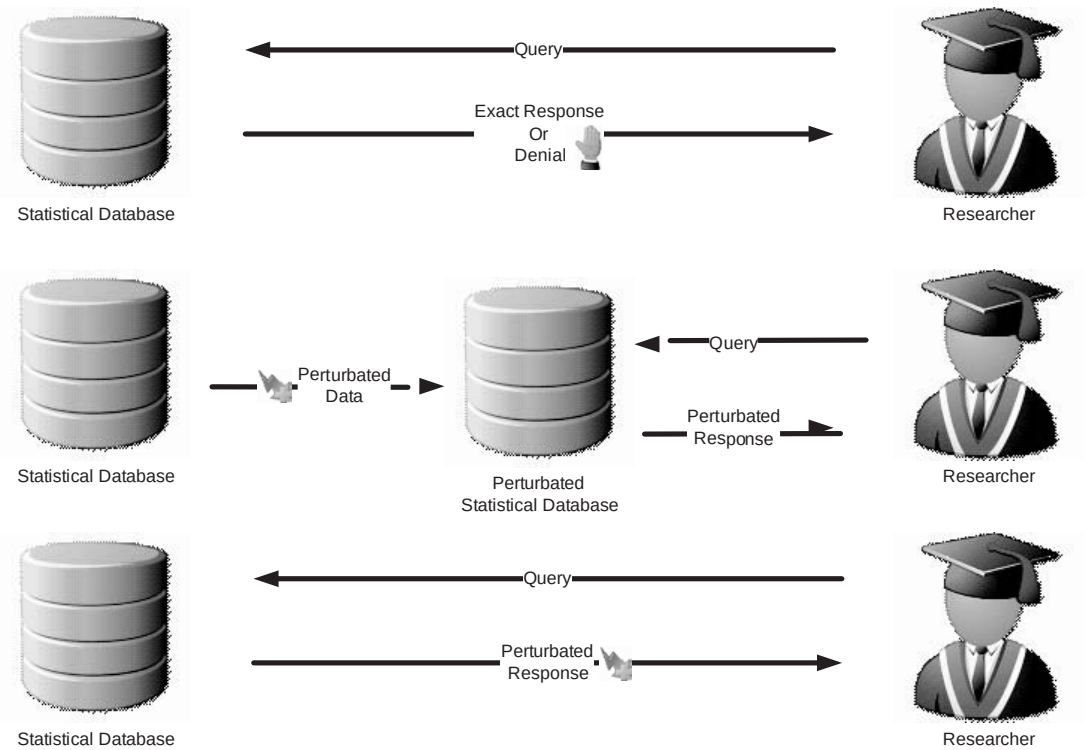
A statistical database contains information about individuals, but allows only aggregate queries (such as asking for the average age instead than Bob Smith's age). Permitting queries that return aggregate results only may seem sufficient to protect an individual's privacy. There is, however, a new problem compared to traditional database security: Inference can be used to infer some secret information.

A very simple example illustrates how inference causes information leakage: If I know Alice is the oldest employee, I can ask "How many employees are older than X years?" Repeating this process for different values of X until the database returns the value 1 allows us to infer Alice's age.

The first approach to prevent this kind of "attack" is to enforce that each query must return data aggregated from at least N rows, with N being certainly larger than 1 and—in the best case—a very large number. Yet, unfortunately, this is no real solution. The first step is to repeatedly ask "How many employees are older than X?" until the system rejects the query because the query would return less than N rows. Now one has identified a set of N+1 employees, including Alice, who are older than X; let X=66 at this point. Next, ask "Tell me the sum of ages of all employees who are older than X?" Let result be R. Next, ask "Tell me the sum of ages of all employees who are not called Alice and are older than X?" Let result be RR. Finally, subtract RR from R to obtain Alice's age.

For an in-depth description we recommend (Castano, 1996).

Figure 1. Security control models (Adam, 1989). Restriction-based protection either gives the correct answer or no answer to a query (top); data may be modified (perturbated) before it is stored in the data warehouse or the statistical database (middle); online perturbation modifies the answers for each query (bottom).



Restriction-Based Techniques

The technique will protect against the aforementioned inference attacks by restricting queries that could reveal confidential data on individuals (Castano, 1996).

Query Set Size Control

Enforcing a minimum set size for returned information does not offer adequate protection for information as we explained in Section 3's introductory example. Denning (Denning, 1982) described trackers that are sequences of queries that are all within the size limits allowed by the database; when combined with AND statements and negations, information on individuals can be inferred. While simple trackers require some background information (for instance, Alice is the only female employee in department D1 who is older than 35), general trackers (Schlörer, 1980), (Denning, 1979) can be used without in-depth background knowledge.

An Audit-Based Expanded Query Set Size Control

The general idea of this control is to store an "assumed information base," that is, to keep a history of all the requests issued by the user. It is also referred to as query set overlap control (Adam, 1989). The system has to calculate all possible inferences (= assumed information base) that could be created based on all queries the user issued; for each new query it has to decide whether the query could be combined with the assumed information base to infer information that should remain confidential.

Perturbation-Based Techniques

Perturbation-based techniques are characterized by modifying the data so that privacy of individuals can still be guaranteed even if more detailed data is returned than in restriction-based techniques. Data can be modified in the original data or in the results returned.

Data Swapping

Data is exchanged between different records so that no original data remains but in a way that the calculated statistics are not impacted.

Random Sample Queries

Set of answers to a specific query are created dynamically so that not all relevant data items are included in the answer. Instead, a random subset is returned. Since the user cannot decide how this random sample is drawn, inference attacks are much harder. If issuing similar queries, the attacker can attempt to remove the sampling errors. These attacks are possible for small data sets; large data sets can be adequately protected by using random sample queries.

Fixed Perturbation (Modify Data)

Unlike the random sample query approach, data modifications and selections are not performed dynamically for each query. Instead, data is modified (though not swapped) as soon as it is stored in the database. The modifications are performed in such a way that they do not significantly influence statistical results.

Query-Based Perturbation

In contrast to fixed perturbation, query-based perturbation modifies data—as the name suggests—for each query dynamically. The advantage is that the amount of perturbation, and thus the accuracy, can be varied individually for different users. More trustworthy users can receive more precise results.

FUTURE TRENDS

According to Mukherjee et al. (2006), the problem with perturbation techniques is that Euclidean-distance-based mining algorithms no longer work well, i.e. distances between data points cannot be reconstructed. The authors propose to use Fourier-related transforms to obscure sensitive data which helps to preserve the original distances at least partly.

Another approach proposed by Domingo-Ferrer et al (2006) is to aggregate data from the original database in small groups. Since this aggregation is done prior to publishing the data the data protector can decide how large such a "micro" group should be.

CONCLUSION

In the previous section we gave a detailed introduction to access control and focused on role-based access control. RBAC is currently the dominant access control model for database systems. While access control is certainly one of the most fundamental mechanisms to ensure that security requirements, such as, confidentiality, integrity, and—at least to some extent—availability are implemented, there are several other aspects that need to be considered, too.

Figure 2 shows how data is extracted from the source databases, transformed and loaded into the data warehouse. It may then be used for analysis and reporting. The transactional database can be secured with the "classical" models of database security such as RBAC or even mandatory access control. Research in this area dates back to the early times of (military) databases in the 1960s. Once data is extracted, transformed and loaded into a DWH, the data will be used in data analysis–this is what a DWH is created for in the first place. DWH security and methods of statistical database security are then used to secure the DWH against attacks such as inference. Nonetheless, overall security can be achieved only by securing all possible attack vectors and not only the operational database (source data) and the DWH itself.

It is essential to secure all of the servers including remote, file, and physical access and to thoroughly understand all communication and transformation processes. Attacks could be launched before or after data is transformed, when it is stored in the data warehouse, or when being retrieved.

REFERENCES

Adam, N. R. , Worthmann, J. C. (1989). Security-control methods for statistical databases: a comparative study. *ACM Comput. Surv.*, 21, 4, 515-556. Retrieved December, 1989, from http://doi.acm.org/10.1145/76894.76895

Bell, D., Padula, L. L.(1975). *Secure Computer System: Unified Exposition and Multics interpretation*. The MITRE Corporation.

Bertino, E.; Latifur R.K.; Sandhu, R. & Thuraisingham, B., Secure Knowledge Management: Confidentiality, Trust, and Privacy. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 2006, 36, 429-438

Böck, B., Klemen, M., Weippl, E.R. (2007). Social Engineering, accepted for publication in: *The Handbook of Computer Networks*.

Castano, S., Martella, G., Samarati, P., Fugini, M. (1994). *Database Security*. Addison-Wesley.

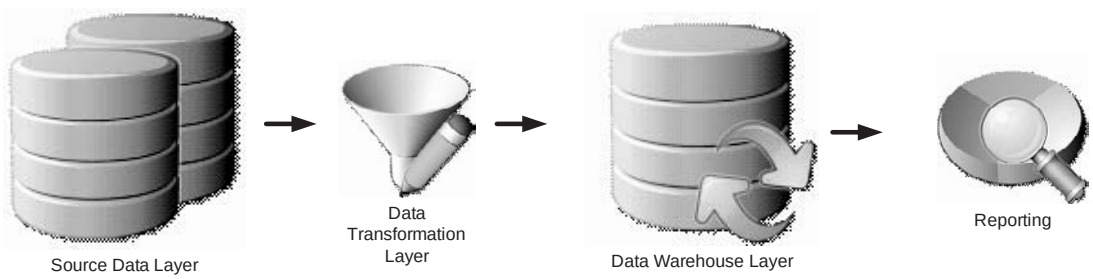
Denning, D. E., Denning, P. J. (1979). The tracker: a threat to statistical database security. *ACM Transactions on Database Systems* 4(1), 76-96. Retrieved March, 1979, from http://doi.acm.org/10.1145/320064.320069

Denning (1982). *Cryptography and Data Security*. Addison-Wesley.

Domingo-Ferrer, J.; Martinez-Balleste, A.; Mateo-Sanz, J. M. & Sebe, F., Efficient multivariate data-oriented microaggregation. *The VLDB Journal*, 2006, 15, 355-369

Mukherjee, S.; Chen, Z. & Gangopadhyay, A. (2006). A privacy-preserving technique for Euclidean dis-

Figure 2. Generic architecture of data flows to a data warehouse



tance-based mining algorithms using Fourier-related transforms. *The VLDB Journal*, 15, 293–315.

Park, J. & Sandhu, R., The UCONABC usage control model. *ACM Transactions on Information Security*, ACM Press, 2004, 7, 128-174

Sandhu, R., Ferraiolo, D., Kuhn, R. (2000). The NIST Model for Role-based Access Control: Towards a Unified Standard. *Proceedings of 5th ACM Workshop on Role-Based Access Control*, 47-63.

Schlörer. (1980, December). Disclosure from statistical databases: Quantitative Aspects of Trackers. *ACM Transactions on Database Systems*, 5(4).

Sandhu, R. S., Coyne, E. J.,Feinstein, H. L., Youman, C. E. (1996). Role-based access control models. *IEEE Computer*, 29(2), 38– 47. Retrieved February, 1996, from http://csdl.computer.org/comp/mags/co/1996/02/r2toc.htm

Sandhu, R.S., Ferraiolo, D., Kuhn, R. (2000, July). The nist model for role-based access control: Towards a unified standard. *Proc. of 5th ACM Workshop on Role-Based Access Control*.

Thuraisingham, B., Data Mining, National Security, Privacy and Civil Liberties, *SIGKDD Explorations*. Volume 4, Issue 2, 2002, 4, 1-5

Thuraisingham, B. (2005). Privacy constraint processing in a privacy-enhanced database management system. *Data & Knowledge Engineering*, 55, 159-188.

Osborn, S., Sandhu, R.S., Munawer, Q. (2000). Configuring role-based access control to enforce mandatory and discretionary access control policies. *ACM Transaction on Information and System Security*, 3(2),85–206.

KEY TERMS

Availability: A system or service is available to authorized users.

CIA: Confidentiality, integrity, availability; the most basic security requirements.

Confidentiality: Only authorized subjects should have read access to information.

DWH: Data warehouse

ETL: The process of Extracting, Transforming (or Transporting) and Loading source data into a DWH.

Integrity: No unauthorized modifications or modifications by unauthorized subjects are made.

OLAP: Online Analytical Processing

Data-Driven Revision of Decision Models

Martin Žnidaršič
Jožef Stefan Institute, Slovenia

Marko Bohanec
Jožef Stefan Institute, Slovenia

Blaž Zupan
University of Ljubljana, Slovenia, and Baylor College of Medicine, USA

INTRODUCTION

Computer models are representations of problem environment that facilitate analysis with high computing power and representation capabilities. They can be either inferred from the data using data mining techniques or designed manually by experts according to their knowledge and experience. When models represent environments that change over time, they must be properly updated or periodically rebuilt to remain useful. The latter is required when changes in the modelled environment are substantial. When changes are slight, models can be merely adapted by revision.

Model revision is a process that gathers knowledge about changes in the modelled environment and updates the model accordingly. When performed manually, this process is demanding, expensive and time consuming. However, it can be automated to some extent if current data about the modelled phenomena is available. Data-based revision is a procedure of changing the model so as to better comply with new empirical data, but which at the same time keeps as much of the original contents as possible. In the following we describe the model revision principles in general and then focus on a solution for a specific type of models, the qualitative multi-attribute decision models as used in DEX methodology.

BACKGROUND

The task of data-driven revision is to adapt the initial model to accommodate for the new data, while at the same time making use of the background knowledge, which was used in the construction of the initial model. Revision is to be applied only when the changes of the modelled concepts are not substantial, that is, if we deal with concept drift (Tsymbal, 2004). If the changes of the modeled system are substantial, it is usually better to construct the model from scratch.

Depending on the field of research, procedures of this kind are most often referred to as knowledge refinement or theory revision. Most of research in this field is done on propositional rule bases (Ginsberg, 1989; Mahoney & Mooney, 1994; Yang, Parekh, Honavar & Dobbs, 1999; Carbonara & Sleeman, 1999) and Bayesian networks (Buntine, 1990; Ramachandran & Mooney, 1998), but many principles of these methods are shared with those of revision procedures for other knowledge representations, such as case-based reasoning systems (Kelbassa, 2003).

Multi-criteria decision models (MCDM) are models used in decision analysis (Clemen, 1996). Data-driven revision can be a valuable tool for the ones that are used for longer periods of time. In our previous work, we have developed revision methods (Žnidaršič & Bohanec, 2005; Žnidaršič, Bohanec & Zupan, 2006) for two types of MCDM models of DEX methodology (Bohanec & Rajkovič, 1990; Bohanec, 2003). An input to MCDM models is criteria-based description of alternatives, where a model represents a utility function to evaluate the given set of criteria values. MCDM models are used for evaluation and analysis of decision alternatives. In contrast to traditional numerical MCDM (Saaty, 1980; Keeney & Raiffa, 1993; Triantaphyllou, 2000), the models of DEX methodology are qualitative and have utility functions in form of if-then rules. The concepts in these models are structured hierarchically and their values are defined according to the values of their immediate descendants in the hierarchy (see Figure 1). This dependency is specified with qualitative rule-based utility functions, which can be defined

as crisp or probabilistic and are usually represented in tabular form (see Table 1). The concepts at the bottom of hierarchy serve as inputs, represent the criteria-based description of alternatives and must be provided by the user.

Models of DEX methodology are usually constructed manually in collaboration of decision analysts and problem domain experts. As an alternative, a method called HINT (Bohanec & Zupan, 2004) was proposed that can infer DEX-like models from data. HINT often requires a large quantity of data and its discovery process may benefit from an active involvement of an expert. The task of revision is simpler and can be achieved completely autonomously with very limited amount of new evidence.

MAIN FOCUS

Revision Goals

For any kind of knowledge representation or model (M), the goal of revision methods (r) is to adapt the model with respect to new evidence from the changed environment. In the case of data-driven revision, evidence is in the form of a set of data items ($D=\{d_1, d_2, .., d_n\}$). The success of adaptation may be demonstrated through the increase of some selected measure (m) that assesses the performance of the model given a set of test data. While standard measures of this kind are, for instance, classification accuracy and mean squared error (Hand, Mannila & Smyth, 2001), any other measure that fits the modelling methodology and problem may be used.

The quality of the model after the revision should thus increase:

$$m\left(r\left(M,D\right),D\right)\geq m\left(M,D\right),$$

where it aims to maximize: $m\left(r\left(M,D\right),D\right)$.

However, revision is a process that tries to preserve the initial background knowledge, rather than subject it entirely to the new data. The maximization from the latter equation must be therefore limited by the type and degree of changes that the revision method is allowed to make. If there exists background data that was used in the initial model construction (D_b) or we are able to reproduce it, we can also limit the revision by fully considering the prior data:

$$m\left(r\left(M,D\right),D_b\right)\geq m\left(M,D_b\right),$$

or at least by minimizing the difference: $m\left(M,D_b\right)-m\left(r\left(M,D\right),D_b\right)$.

Figure 1. A simple hierarchy of concepts of a decision model for car purchase

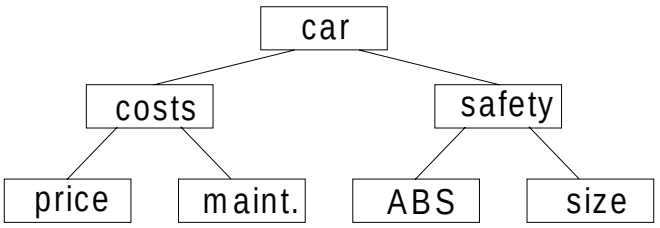


Table 1. Utility function of the concept safety

ABS	size	safety
no	small	<v.good:0.0, good:0.0, accep.:0.1, bad:0.9>
no	medium	<v.good:0.0, good:0.1, accep.:0.7, bad:0.2>
no	big	<v.good:0.1, good:0.8, accep.:0.1, bad:0.0>
yes	small	<v.good:0.0, good:0.0, accep.:0.3, bad:0.7>
yes	medium	<v.good:0.1, good:0.6, accep.:0.3, bad:0.0>
yes	big	<v.good:0.8, good:0.2, accep.:0.0, bad:0.0>

In any case, we must prevent excessive adaptation of a model to the learning data set. This phenomenon is called over-fitting and is a well known problem in data mining. Usually it is controlled by splitting data to separate data sets for learning and testing purposes. There are many approaches to that (Hand, Mannila & Smyth, 2001), but ten fold cross-validation of data is the most common one. With regard to revision, in addition to limiting the type and degree of changes, it is also important to assess the model performance measure on a partition of data that was not used in the learning (revision) process.

Data Processing

New data items get processed by revision methods in two most common ways:

1. *Incremental (singleton iterations)*: As it is beneficial for revision methods to be able to operate even with a single piece of evidence, they are usually implemented as iterations of a basic step: revision based on a single piece of evidence. This way, the revision method considers one data item at a time and performs revision accordingly. However, iterative revision is problematic as the result depends on the order of the data items. To circumvent this problem, the data items are usually ordered in one of the following ways:
   a. *Temporal ordering*: This approach is appropriate when data items are obtained from a data stream or have time stamps, so that they are processed in first-in-first-out manner. Such a setting also makes easy to observe and follow the concept drift.
   b. *Relevance ordering*: Sometimes we can order data items by relevance that is either defined manually in the data acquisition phase or defined through some quantitative measure of relevance. For example, we could order data by observing the quality of model prediction on each data instance, that is, ranking the instances by $m(M, d_i)$, and first revise the model with the data that have the highest or the lowest corresponding value.
   c. *Random ordering*: Random ordering is not informative ordering, but it is often used when the data cannot be ordered in any informative way. This ordering is used also for purposes of method evaluation with unordered data sets. In such situations, the batch approach (see below) is more appropriate. If random ordering is used in singleton iterations of revision, we must take care for the reproducibility of results by saving the learning data set in order of processing or setting and saving the random seed.
2. *Batch processing*: Some revision methods use a whole data set in one step, in a batch (Hand, Mannila & Smyth, 2001). This can be done by utilizing the statistical features of a batch, such as frequencies, medians or averages of attribute values. Batch processing can be also simulated with algorithms for iterative revision if we save the proposed revision changes for each data item and apply them averaged after processing the last data item from the learning data set. Batch methods tend to be less prone to unwanted deviations of models, as the influence of single data items is restrained. Another benefit is easy reproducibility of experimental results, since data properties that are used in batch processing do not change with data item perturbations.

Challenges

The developer of revision methods usually meets the following set of problems:

1. *Unwanted model deviation*: In order to increase the performance of the model on new data, revision makes changes to the model. These changes can cause the model to deviate in an unwanted way, which usually happens when the impact of new data is overemphasized and/or the changes are not properly controlled to stay in line with the original model. This is especially problematic in the revision of decision models, where elements of the model consist of predefined set of relations with their intrinsic interpretation. For the models in form of neural networks, for example, a deviation cannot be regarded as unwanted, as long as it increases the performance measure. Unwanted model deviation can be limited by controlling the changes to be allowed to happen only near the original resolution path of a data item. Another approach would be to check every proposed change for compatibility with the behavior of

the original model on data set that was used to build it, the new data set, or both. This implies continuous computing of model performance with or without a change on at least one dataset and often causes the next potential problem of revision methods.

2. *Computational complexity*: As the task of model revision is simpler than that of model construction from the scratch, we expect its algorithms to be also less computationally demanding. However, the advantage of this simplicity is more often compensated with very low data demand than with the decrease of computational load. Unlike the model construction, revision is a process that is repeated, potentially in real time. It is thus essential to adapt the revision method (for instance the unwanted deviation limitation techniques) to the model and data size. The adaptation is also needed to exploit the features of the problem and the modelling methodology in order to limit the search space of solutions.
3. *Heterogeneous models*: The knowledge contained in the models can be heterogeneous in the sense that some of it was initially supported by more evidence, e.g., the user that defined it was more confident than in the other parts of the model. Not all parts of the model should therefore be revised to the same extent, and the part that relies on the more solid background knowledge should change less in the revision. To take this heterogeneity into account, we must make it explicit in the models. This information is often provided in form of parameters that describe the higher-order uncertainty (Lehner, Laskey, & Dubois, 1996; Wang 2001), i.e. the confidence about the knowledge provided in the model (values, probabilities, etc.). There are data-based revision methods that can utilize such information (Wang, 1993; Wang, 2001; Žnidaršič, Bohanec & Zupan, 2006). An example of one such method is presented in the next section.

Solutions for MCDM

Data-based revision of MCDM is both interesting and challenging. We have developed and experimentally tested two distinct revision methods for two variants of qualitative MCDM of DEX methodology. Both methods operate on probabilistic utility functions (see Table 1) and both rely on user defined parameters of the degree of changes. The first method (*rev_trust*) uses a parameter named *trust* that is provided for the whole learning data set and represents the users' trust in new empirical data. The second method (*rev_conf*) is intended to revise models containing heterogeneous knowledge. *Rev_conf* can exploit the parameters named *confidence*, which are provided for each rule in every utility function of the model to represent the users' confidence about the values in these rules.

The two methods have a common incremental operation strategy: they process data iteratively by considering one data item at a time. The data item consists of values of the input attributes (concepts at the bottom of hierarchy) and a value of the topmost concept. We represent a generic data item with a set of values $d_1, \ldots, d_n$ of input concepts and with the value of the topmost target concept g that also represents the utility of the item:

$[d_1, \ldots, d_n, g]$.

For the model from Figure 1, the data item could be for instance: [price=low, maint.=medium, ABS=yes, size=small, car=bad]. Both methods proceed through the hierarchy with each data item in a recursive top-down fashion. The revision first changes the utility function of the topmost concept (G), for which a true value (g) is given. The change is applied to the probability distribution in the rule with the highest weight according to the input values. The change is made so as to emphasize the true value g of its goal attribute G for the given values of input concepts. In the second step, the algorithm investigates whether the probability of g can be increased for current alternative without changing the utility function of G, but rather by changing the utility functions of the G's descendants. A combination of descendants values, which results in such a change, is then used as the true goal value in the next revision step, when this process is repeated on each descendant.

Apart from the described similar principles of operation, the methods *rev_trust* and *rev_conf* differ in some important aspects of the above mentioned procedures. The differences are briefly presented here and summarized in Table 2. For more elaborate descriptions of algorithms see the literature (Žnidaršič & Bohanec, 2005; Žnidaršič, Bohanec & Zupan, 2006).

Table 2: Summary of differences between the methods rev_trust and rev_conf

procedure	*rev_trust*	*rev_conf*
revision change	$p(g)$ + *trust*, normalization	according to the *confidence* of goal distribution
selection of descendant's value	values yielding distribution closest to target distribution biased towards g	values yielding max. value of g
divergence control	measure check	near resolution path

The first difference is in the computation of revision change. *Rev_trust* increases the probability of g in the selected goal distribution by a user defined parameter *trust* and then normalizes the distribution, whereas *rev_conf* uses the parameter *confidence* of the selected distribution and increases the g's probability accordingly (higher *confidence* - smaller revision).

The next difference is in the selection of candidate values for revision in descendants. *Rev_trust* selects a specified number of value combinations, which result closest to a special precomputed goal distribution biased towards g. *Rev_conf* is simpler in this aspect: it selects a value combination that results in a distribution with maximum probability of g. If there are more such combinations, all get selected.

The last difference is associated with the previous one, namely, as both methods potentially select more than one candidate combination of descendants values for the next recursive step, only the best one must be chosen. This is important for avoiding unwanted model deviations. *Rev_trust* selects the best combination according to a performance measure, whereas *rev_conf* selects the combination, which is the closest to the original resolution path of the input values.

Experimental results indicate (Žnidaršič & Bohanec, 2005; Žnidaršič, Bohanec & Zupan, 2006) that *rev_trust* is more successful in controlling the unwanted divergence and in increasing the performance measure. However, the second approach is much faster and offers heterogeneous revision of models. *Rev_conf* is also more autonomous, as it automatically adapts the rate of revision once the *confidence* parameters in the model are set. Therefore it is not easy to recommend one approach over the other. There is also room for mixed approaches with a subset of features from the first method and a subset of features from the other. Investigating the most interesting mixture of approaches from the both presented methods is the topic of further research. The most promising seems to be the narrowing of the space of revision options near the resolution path, but with a bigger set of candidates, among which the best ones are pinpointed using a performance measure check. Such a check proved to be beneficial for avoiding unwanted diversion, while at the same time its computational demands should not be problematic on a narrowed amount of candidate changes.

FUTURE TRENDS

With growing availability of data and limited amount of time for development and maintenance of computer models, automatic revision is a valuable approach for extending their lifetime. In the future, the revision methods could be adapted also to representations of higher complexity, like mechanistic models, which are commonly used in natural sciences. With advances in sensor and monitoring technology, the environmental management is one among many domains that could largely benefit from revision methods for complex phenomena representations.

CONCLUSION

Autonomous data-driven revision is a useful maintenance tool, which aims at saving valuable expert time for the adaptation of computer models to slight changes in the modelled environment. Revision methods generally share some common goals and features, but there is a wide choice of approaches even inside a particular modelling methodology. Their choice depends on the methodology, data and/or model size and the specific features of the modelling problem. Knowing the specific problem and general features of revision approaches, we must decide whether we should build the model from the scratch or should we revise it and how should we proceed to revise it. For

multi-criteria decision models of DEX methodology, we have introduced two alternative revision methods, *rev_trust* and *rev_conf*. The former is useful for small and homogeneous models, whereas the latter is more appropriate for large and heterogenous models.

REFERENCES

Bohanec, M., & Rajkovič, V. (1990). DEX : An Expert System Shell for Decision Support. *Sistemica, 1*(1), 145-157.

Bohanec, M. (2003). Decision support. In: Mladenić, D., Lavrač, N., Bohanec, M. & Moyle, S. (Eds.), *Data mining and decision support: Integration and collaboration* (pp. 23-35). Kluwer Academic Publishers.

Bohanec, M., & Zupan, B. (2004). A function-decomposition method for development of hierarchical multi-attribute decision models. *Decision Support Systems, 36*(3), 215-233.

Buntine, W. (1991). Theory refinement of Bayesian networks. In D'Ambrosio, B., Smets, P., & Bonissone, P. (Eds.), *Uncertainty in Artificial Intelligence: Proceedings of the Seventh Conference* (pp. 52-60). Los Angeles, California.

Carbonara, L., & Sleeman, D. H. (1999). Effective and efficient knowledge base refinement. *Machine Learning, 37*(2), 143-181.

Clemen, R. T. (1996). *Making Hard Decisions: An Introduction to Decision Analysis*. Duxbury Press, Pacific Grove, CA.

Ginsberg, A. (1989). Knowledge base refinement and theory revision. In *Proceedings of the Sixth International Workshop of Machine Learning* (pp. 260-265).

Hand, D., Mannila, H., & Smyth, P. (2001). *Principles of Data Mining*. The MIT Press.

Keeney, R. L., Raiffa, H. (1993). *Decisions with Multiple Objectives*. Cambridge University Press.

Kelbassa. H. W. (2003). Optimal Case-Based Refinement of Adaptation Rule Bases for Engineering Design. In Ashley, K. D. & Bridge, D. G. (Eds.), *Case-Based Reasoning Research and Development, 5th International Conference on Case-Based Reasoning* (pp. 201-215). Trondheim, Norway: Lecture Notes in Computer Science Vol. 2689, Springer.

Lehner, P. E., Laskey, K. B., & Dubois, D. (1996). An introduction to issues in higher order uncertainty. *IEEE Transactions on Systems, Man and Cybernetics, A, 26*(3), 289-293.

Mahoney, M. J., & Mooney, R. J. (1994). Comparing methods for refining certainty-factor rule-bases. In Cohen, W. W. & Hirsh, H. (Eds.), *Proceedings of the Eleventh International Conference of Machine Learning* (pp. 173-180). New Brunswick, NJ. Los Altos, CA: Morgan Kaufmann.

Ramachandran, S., & Mooney, R. J. (1998). Theory refinement for Bayesian networks with hidden variables. In *Proceedings of the International Conference on Machine Learning* (pp. 454-462). Madison, WI: Morgan Kaufman.

Saaty, T. L. (1980). *The Analytic Hierarchy Process*. McGraw-Hill.

Triantaphyllou, E. (2000). *Multi-Criteria Decision Making Methods: A Comparative Study*. Kluwer Academic Publishers.

Tsymbal, A. (2004). *The Problem of Concept Drift: Definitions and Related Work* (Tech. Rep. TCD-CS-2004-14). Ireland: Trinity College Dublin, Department of Computer Science.

Wang, P. (1993). Belief Revision in Probability Theory. In *Proceedings of the Ninth Annual Conference on Uncertainty in Artificial Intelligence* (pp. 519-526). Washington, DC, USA: Morgan Kaufmann Publishers.

Wang, P. (2001). Confidence as higher-order uncertainty. In *Proceedings of the Second International Symposium on Imprecise Probabilities and Their Applications* (pp. 352-361).

Yang, J., Parekh, R., Honavar, V. & Dobbs, D. (1999). Data-driven theory refinement using KBDistAl. In *Proceedings of the Third Symposium on Intelligent Data Analysis* (pp. 331-342). Amsterdam, The Netherlands.

Žnidaršič, M. & Bohanec, M. (2005). Data-based revision of probability distributions in qualitative multi-attribute decision models. *Intelligent Data Analysis, 9*(2), 159-174.

Žnidaršič, M., Bohanec, M. & Zupan, B. (2006). Higher-Order Uncertainty Approach to Revision of Probabilistic Qualitative Multi-Attribute Decision Models. In *Proceedings of the International Conference on Creativity and Innovation in Decision Making and Decision Support (CIDMDS 2006)*. London, United Kingdom.

KEY TERM

Background Knowledge: Knowledge used in a process, such as a construction of computer models, which is not made explicit. It influences the knowledge representations indirectly, for example by defining a structure of concepts or a set of values that a concept might have.

Batch Processing: Processing data in a batch at a time. Usually by considering statistical properties of a data set, like averages and frequencies of values.

Concept Drift: A change of concepts in the modelled environment. It causes the models, which are based on old empirical data, to become inconsistent with the new situation in the environment and new empirical data.

Data-Driven Revision: A process of revising knowledge and knowledge representations with empirical data, which describes the current situation in the modelled environment. Revision takes the initial state of knowledge as a starting point and preserves it as much as possible.

Incremental Processing: Processing data one data item at a time. An approach followed when processing must be possible to perform even on a single data item and when a suitable ordering of data items exists.

Mechanistic Model: A computer model, which is built so as to represent particular natural phenomena in the highest possible detail. Mechanistic models are based on domain expert knowledge and general physical laws, hence the name mechanistic or physical. They are complex and computationally demanding, but very useful for detailed simulations of processes which would be hard to empirically experiment with (such as big-scale nuclear tests).

Model Deviation: A change of parameters in the model, which causes concept representations to become inconsistent with their original meaning. Usually this is a consequence of uncontrolled adaptation of the model to empirical data.

Multi Criteria Decision Models (MCDM): Decision models for decision problems, which aim at satisfying several (usually conflicting) criteria. In these models, the main decision goal is decomposed into several hierarchical layers of criteria. This way, the problem is decomposed into smaller and smaller subproblems in order to become feasible to accurately describe and analyze.

Decision Tree Induction

Roberta Siciliano
University of Naples, Federico II, Italy

Claudio Conversano
University of Cagliari, Italy

INTRODUCTION

Decision Tree Induction (DTI) is a tool to induce a classification or regression model from (usually large) datasets characterized by n objects (records), each one containing a set **x** of numerical or nominal attributes, and a special feature *y* designed as its outcome. Statisticians use the terms "predictors" to identify attributes and "response variable" for the outcome. DTI builds a model that summarizes the underlying relationships between **x** and *y*. Actually, two kinds of model can be estimated using decision trees: *classification trees* if *y* is nominal, and *regression trees* if *y* is numerical. Hereinafter we refer to classification trees to show the main features of DTI. For a detailed insight into the characteristics of regression trees see Hastie et al. (2001).

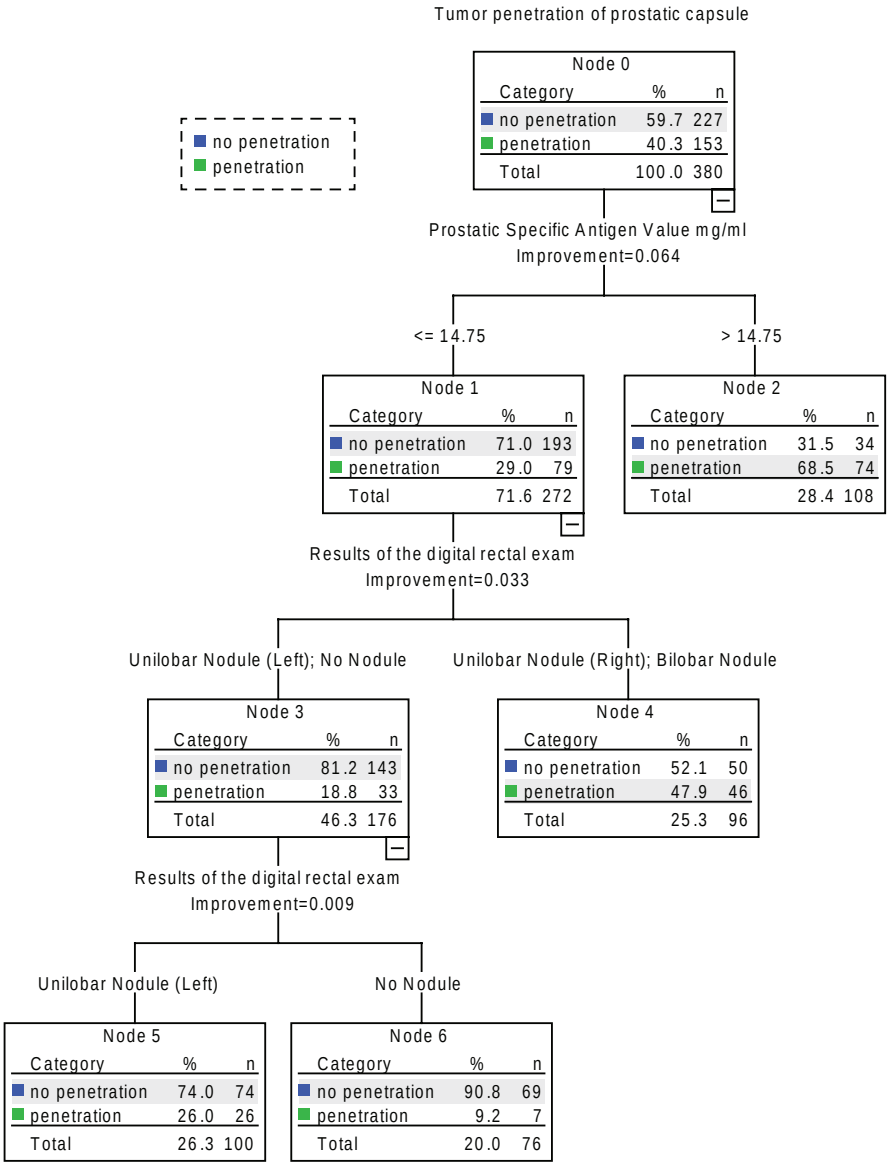
As an example of classification tree, let us consider a sample of patients with prostate cancer on which data such as those summarized in Figure 1 have been collected. Suppose a new patient is observed and we want to determine if the tumor has penetrated the prostatic capsule on the basis of the other available information. Posing a series of questions about the characteristic of the patient can help to predict the tumor's penetration. DTI proceeds in such a way, inducing a series of follow-up (usually binary) questions about the attributes of an unknown instance until a conclusion about what is its most likely class label is reached. Questions and their alternative answers can be represented hierarchically in the form of a decision tree, such as the one depicted in Figure 2.

The decision tree contains a root node and some internal and terminal nodes. The root node and the internal ones are used to partition instances of the dataset into smaller subsets of relatively homogeneous classes. To classify a previously unlabelled instance, say $i^* \left(i^* = 1, \ldots, n\right)$, we start from the test condition in

Figure 1. The prostate cancer dataset

Age in years	Result of the digital rectal exam	Result of he detection of capsular involvment in rectal exam	Prostatic specific antigen value mg/ml	Tumor penetration of prostatic capsule
65	Unilobar Nodule (Left)	No	1.400	*no penetration*
70	No Nodule	Yes	4.900	*no penetration*
71	Unilobar Nodule (Right)	Yes	3.300	*penetration*
68	Bilobar Nodule	Yes	31.900	*no penetration*
69	No Nodule	No	3.900	*no penetration*
68	No Nodule	Yes	13.000	*no penetration*
68	Bilobar Nodule	Yes	4.000	*penetration*
72	Unilobar Nodule (Left)	Yes	21.200	*penetration*
72	Bilobar Nodule	Yes	22.700	*penetration*
…	……..	…..	…….	……..…

Figure 2. An illustrative example of a decision tree for the prostate cancer classification



the root node and follow the appropriate pattern based on the outcome of the test. When an internal node is reached a new test condition is applied, and so on down to a terminal node. Encountering a terminal node, the modal class of the instances of that node is the class label of i^*. Going back to the prostate cancer classification problem, a new subject presenting a prostatic specific antigen value lower than 4.75, and an unilobar nodule on the left side will be classified as "no penetration". It is evident that decision trees can easily be converted into IF-THEN rules and used for decision making purposes.

BACKGROUND

DTI is useful for data mining applications because of the possibility to represent functions of numerical and

nominal attributes as well as of its feasibility, predictive ability and interpretability. It can effectively handle missing values and noisy data and can be used either as an explanatory tool for distinguishing objects of different classes or as a prediction tool to class labels of previously unseen objects.

Some of the well-known DTI algorithms include ID3 (Quinlan, 1983), CART (Breiman et al., 1984), C4.5 (Quinlan, 1993), FAST (Mola and Siciliano, 1997), SLIQ (Metha et al., 1997) and GUIDE (Loh, 2002). All these algorithms use a greedy, top-down recursive partitioning approach. They primarily differ in terms of the splitting criteria, the type of splits (2-way or multiway) and the handling of the overfitting problem.

MAIN THRUST

DTI uses a greedy, top-down recursive partitioning approach to induce a decision tree from data. In general, DTI involves the following tasks: decision tree growing and decision tree pruning.

Growing a Decision Tree

Decision trees use a greedy heuristic to make a series of locally optimum decisions about which attribute value to use for data partitioning. A test condition depending on a splitting method is applied to partition the data into more homogeneous subgroups at each step of the greedy algorithm.

Splitting methods differ with respect to the type of attribute: for nominal attributes the test condition is expressed as a question about one or more of its values $(\text{e.g.: } x_i = a?)$ whose outcomes are "Yes"/"No". Grouping of attribute values is required for algorithms using 2-way splits. For ordinal or continuous attributes the test condition is expressed on the basis of a threshold value υ such as $(x_i \le \upsilon?)$ or $(x_i > \upsilon?)$. Considering all possible split points υ, the best one υ^* partitioning the instances into homogeneous subgroups is selected. A splitting function accounts for the class distribution of instances both in the parent node and in the children nodes computing the decrease in the degree of impurity $\Delta(\upsilon, t)$ for each possible υ. It depends on the impurity measure of the parent node t (prior to splitting) against the weighted-averaged impurity measure of the children nodes (after splitting), denoted with t_r and t_l.

As for the single node t, if $p(i \mid t)$ is the proportion of instances belonging to class i and C classes are observed, the impurity $\omega(t)$ can be computed alternatively as follows:

Entropy: $\omega(t) = -\sum_{i=1}^{C} p(i \mid t) \log p(i \mid t)$

Gini: $\omega(t) = 1 - \sum_{i=1}^{C} \left[p(i \mid t) \right]^2$

Classification Error: $\omega(t) = 1 - \max_i \left[p(i \mid t) \right]$

To select the best split point, the value υ^* such that $\Delta(v,t) = \omega(t) - \left[\omega(t_l) p(t_l) + \omega(t_r) p(t_r) \right] = \max!$ with $p(t_l)$ and $p(t_r)$ are the proportions of instances falling into the two sub-nodes, where has to be found.

When entropy is used as the impurity measure $\omega(t)$, the degree of impurity $\Delta(\upsilon, t)$ is known as Information Gain.

Data partitioning proceeds recursively until a stopping rule is satisfied: this usually happens when the number of instances in a node is lower than a previously-specified minimum number necessary for splitting, as well as when the same instances belong to the same class or have the same attribute values.

Pruning a Decision Tree

DTI algorithms require a pruning step following the tree growing one in order to control for the size of the induced model and to avoid in this way data overfitting. Usually, data is partitioned into a training set (containing two-third of the data) and a test set (with the remaining one-third). Training set contains labeled instances and is used for tree growing. It is assumed that the test set contains unlabelled instances and is used for selecting the final decision tree: to check whether a decision tree, say D, is generalizable, it is necessary to evaluate its performance on the test set in terms of misclassification error by comparing the true class labels of the test data against those predicted by D. Reduced-size trees perform poorly on both training and test sets causing underfitting. Instead, increasing the size of D improves both training and test errors up to a "critical size" from which test errors increase even though corresponding training errors decrease. This means that D overfits the data and cannot be generalized to class prediction of unseen objects. In the machine learning framework, the training error is named resbstitution error and the test error is known as *generalization error*.

It is possible to prevent overfitting by haltering tree growing before it becomes too complex (*pre-pruning*). In this framework, one can assume the training data is a good representation of the overall data and use the resubstitution error as an optimistic estimate of the error of the final DTI model (*optimistic approach*). Alternatively, Quinlan (1987) proposed a pessimistic approach that penalizes complicated models by assigning a cost penalty to each terminal node of the decision tree: for C4.5, the generalization error is $\pi(t)/n_t + \varepsilon$, where, for a node t, n_t is the number of instances and $\pi(t)$ is the missclassification error. It is assumed $\pi(t)$ follows a Binomial distribution and that ε is the upper bound for $\pi(t)$ computed from such a distribution (Quinlan, 1993).

An alternative pruning strategy is based on the growing of the entire tree and the subsequent retrospective trimming of some of its internal nodes (*post-pruning*): the subtree departing from each internal node is replaced with a new terminal node whose class label derives from the majority class of instances belonging to the subtree. The subtree is definitively replaced by the terminal node if such a replacement induces an improvement of the generalization error. Pruning stops when no further improvements can be achieved. The generalization error can be estimated through either the optimistic or pessimistic approaches.

Other post-pruning algorithms, such as CART, use a complexity measure that accounts for both tree size and generalization error. Once the entire tree is grown using training instances, a penalty parameter expressing the gain/cost tradeoff for trimming each subtree is used to generate a sequence of pruned trees, and the tree in the sequence presenting the lowest generalization error (0-SE rule) or the one with a generalization error within one standard error of its minimum (1-SE rule) is selected. Cappelli et al. (2002) improved this approach introducing a statistical testing pruning to achieve the most reliable decision rule from a sequence of pruned trees.

Pruning algorithms can be combined with *k-fold cross-validation* when few instances are available. Training data is divided into k disjoint blocks and a tree is grown k times on k-1 blocks estimating the error by testing the model on the remaining block. In this case, the generalization error is the average error made for the k runs.

Averaging Decision Trees

A different approach is based on the generation of a set of candidate trees and their aggregation in order to improve their generalization ability. Tree averaging requires the definition of a suitable set of trees and their associated weights and classifies a new object by averaging over the set of weighted trees (Oliver and Hand, 1995). Either a compromise rule or a consensus rule can be used for averaging. An alternative method consists in summarizing the information of each tree in a table cross-classifying terminal nodes outcomes with the response classes in order to assess the generalization ability through a statistical index and select the tree providing the maximum value of such index (Siciliano, 1998).

Ensemble Methods

Ensemble methods are based on a weighted or non weighted aggregation of single trees (the so called *weak learners*) in order to improve the overall generalization error induced by each single tree. These methods are more accurate than a single tree if they have a generalization error that is lower than random guessing and if the generalization errors of the different trees are uncorrelated (Dietterich, 2000).

Bagging (Bootstrap Aggregating) (Breiman, 1996) works by randomly replicating training instances in order to induce single trees whose aggregation by majority voting provides the final classification. Breiman (1998) argues that bagging is able to improve the performance of unstable classifiers (i.e. trees with high variance). Thus, bagging is said to be a reduction variance method.

AdaBoost (Adaptive Boosting) (Freud and Schapire, 1996; Schapire et al., 1998) uses iteratively bootstrap replication of the training instances. At each iteration, previously-misclassified instances receive higher probability of being sampled. The final classification is obtained by majority voting. Boosting forces the decision tree to learn by its error, and is able to improve the performance of trees with both high bias (such as single-split trees) and variance.

Finally, Random Forest (Breiman, 2001) is an ensemble of unpruned trees obtained by randomly resampling training instances and attributes. The overall performance of the method derives from averaging the generalization errors obtained in each run. Simultane-

ously, suitable measures of attributes importance are obtained to enrich the interpretation of the model.

FUTURE TRENDS

Combining Trees with Other Statistical Tools

A consolidated literature about the incorporation of parametric and nonparametric models into trees appeared in recent years. Several algorithms have been introduced as hybrid or functional trees (Gama, 2004), among the machine learning community. As an example, DTI is used for regression smoothing purposes in Conversano (2002): a novel class of semiparametric models named Generalized Additive Multi-Mixture Models (GAM-MM) is introduced to perform statistical learning for both classification and regression tasks. These models works using an iterative estimation algorithm evaluating the predictive power of a set of suitable mixtures of smoothers/classifiers associated to each attribute. Trees are part of such a set that defines the estimation functions associated to each predictor on the basis of bagged scoring measures taking into account the trade-off between estimation accuracy and model complexity. Other hybrid approaches are presented in Chan and Loh (2004), Su et al. (2004), Choi et al. (2005) and Hothorn et al. (2006).

Another important streamline highlights the use of decision trees for exploratory and decision purposes. Some examples are the multi-budget trees and two-stage discriminant trees (Siciliano et al., 2004), the stepwise model tree induction method (Malerba et al. (2004), the multivariate response trees (Siciliano and Mola, 2000) and the three-way data trees (Tutore et al., 2007). In the same line, Conversano et al. (2001) highlights the idea of using DTI to preliminarily induce homogenous subsamples in data mining applications.

Trees for Data Imputation and Data Validation

DTI has also been used to solve data quality problems because of its extreme flexibility and its ability to easily handle missing data during the tree growing process. But DTI does not permit to perform missing data imputation in a straightforward manner when dealing with multivariate data. Conversano and Siciliano (2004) defined a decision tree-based incremental approach for missing data imputation using the principle of "statistical learning by information retrieval". Data can be used to impute missing values in an incremental manner, starting from the subset of instances presenting the lowest proportion of missingness up to the subset presenting the maximum number of missingness. In each step of the algorithm a decision tree is learned for the attribute presenting missing values using complete instances. Imputation is made for the class attribute inducing the final classification rule through either cross-validation or boosting.

CONCLUSION

In the last two decades, computational enhancements highly contributed to the increase in popularity of DTI algorithms. This cause the successful use of Decision Tree Induction (DTI) using recursive partitioning algorithms in many diverse areas such as radar signal classification, character recognition, remote sensing, medical diagnosis, expert systems, and speech recognition, to name only a few. But recursive partitioning and DTI are two faces of the same medal. While the computational time has been rapidly reducing, the statistician is making more use of computationally intensive methods to find out unbiased and accurate classification rules for unlabelled objects. Nevertheless, DTI cannot result in finding out simply a number (the misclassification error), but also an accurate and interpretable model.

Software enhancements based on interactive user interface and customized routines should empower the effectiveness of trees with respect to interpretability, identification and robustness.

REFERENCES

Breiman, L., Friedman, J.H., Olshen, R.A., & Stone C.J. (1984). *Classification and regression trees*. Wadsworth, Belmont CA.

Breiman, L. (1996). Bagging predictors. *Machine Learning, 24*, 123-140.

Breiman, L. (2001). Random forests. *Machine Learning, 45*, 5-32.

Cappelli, C., Mola, F., & Siciliano, R. (2002). A Statistical approach to growing a reliable honest tree. *Computational Statistics and Data Analysis, 38,* 285-299.

Chan, K. Y., & Loh, W. Y. (2004). LOTUS: An algorithm for building accurate and comprehensible logistic regression trees. *Journal of Computational and Graphical Statistics, 13,* 826–852.

Choi, Y., Ahn, H., & Chen, J.J. (2005). Regression trees for analysis of count data with extra Poisson variation. *Computational Statistics and Data Analysis, 49,* 893–915.

Conversano, C. (2002). Bagged mixture of classifiers using model scoring criteria. *Patterns Analysis & Applications, 5*(4), 351-362.

Conversano, C., Mola, F., & Siciliano, R. (2001). Partitioning algorithms and combined model integration for data mining. *Computational Statistics, 16,* 323-339.

Conversano, C., & Siciliano, R. (2004). *Incremental Tree-based missing data imputation with lexicographic ordering.* Interface 2003 Proceedings, cd-rom.

Dietterich, T.G. (2000) Ensemble methods in machine learning. In J. Kittler and F. Roli, *multiple classifier system.* First International Workshop, MCS 2000,Cagliari, volume 1857 of lecture notes in computer science. Springer-Verlag.

Freund, Y., & Schapire, R. (1996). Experiments with a new boosting algorithm, *Machine Learning: Proceedings of the Thirteenth International Conference,* 148-156.

Gama, J. (2004). "Functional trees". *Machine Learning, 55,* 219–250.

Hastie, T., Friedman, J. H., & Tibshirani, R., (2001). *The elements of statistical learning: Data mining, inference and prediction.* Springer.

Hothorn, T., Hornik, K., & Zeileis, A. (2006). Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical Statistics, 15,* 651–674.

Loh, W.Y. (2002). Regression trees with unbiased variable selection and interaction detection. *Statistica Sinica, 12,* 361-386.

Malerba, D., Esposito, F., Ceci, M., & Appice, A. (2004). Top-down induction of model trees with regression and splitting nodes. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 26, 6.

Mehta, M., Agrawal, R. & Rissanen J. (1996). SLIQ. A fast scalable classifier for data mining. In *Proceeding of the International Conference on Extending Database Technology EDBT,* 18-32.

Mola, F., & Siciliano, R. (1997). A fast splitting algorithm for classification trees. *Statistics and Computing, 7,* 209–216.

Oliver, J.J., & Hand, D. J. (1995). On pruning and averaging decision trees. *Machine Learning: Proceedings of the 12th International Workshop*,430-437.

Quinlan, J.R. (1983). Learning efficient classification procedures and their application to chess and games. In R.S., Michalski, J.G., Carbonell, & T.M., Mitchell (ed.), *Machine Learning: An Artificial Intelligence Approach, 1, Tioga Publishing,* 463-482.

Quinlan, J.R., (1987). Simplifying decision tree. Internat. *J. Man-Machine Studies, 27,* 221–234.

Quinlan, J. R. (1993). *C4.5: Programs for machine learning.* Morgan Kaufmann.

Schapire, R.E., Freund, Y., Barlett, P., & Lee, W.S. (1998). Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics, 26*(5), 1651-1686.

Siciliano, R. (1998). Exploratory versus decision trees. In Payne, R., Green, P. (Eds.), *Proceedings in Computational Statistics.* Physica-Verlag, 113–124.

Siciliano, R., Aria, M., & Conversano, C. (2004). Tree harvest: Methods, software and applications. In J. Antoch (Ed.), *COMPSTAT 2004 Proceedings.* Springer, 1807-1814.

Siciliano, R. & Mola, F. (2000). Multivariate data analysis through classification and regression trees. *Computational Statistics and Data Analysis, 32,* 285-301. Elsevier Science.

Su, X., Wang, M., & Fan, J. (2004). Maximum likelihood regression trees. *Journal of Computational and Graphical Statistics, 13,* 586–598.

Tutore, V.A., Siciliano, R., & Aria, M. (2007). *Conditional classification trees using instrumental variables.* Advances in Intelligent Data Analysis VII, Berthold M.R, Shawe-Taylor J., Lavrac N (eds.) Springer, 163-173.

KEY TERMS

AdaBoost (Adaptive Boosting): An iterative bootstrap replication of the training instances such that, at any iteration, misclassified instances receive higher probability to be included in the current bootstrap sample and the final decision rule is obtained by majority voting.

Bagging (Bootstrap Aggregating): A bootstrap replication of the training instances, each having the same probability to be included in the bootstrap sample. The single decision trees (weak learners) are aggregated to provide a final decision rule consisting in either the average (for regression trees) or the majority voting (for classification trees) of the weak learners' outcome.

Classification Tree: An oriented tree structure obtained by recursively partitioning a sample of objects on the basis of a sequential partitioning of the attribute space such to obtain internally homogenous groups and externally heterogeneous groups of objects with respect to a categorical attribute.

Decision Rule: The result of an induction procedure providing the final assignment of a response class/value to a new object for that only the attribute measurements are known. Such rule can be drawn in the form of a decision tree.

Ensemble: A combination, typically by weighted or unweighted aggregation, of single decision trees able to improve the overall accuracy of any single induction method.

Exploratory Tree: An oriented tree graph formed by internal nodes, that allow to describe the conditional interaction paths between the response attribute and the other attributes, and terminal nodes, that allow to describe the conditional interaction paths between the response attribute and the other attributes.

Fast Algorithm for Splitting Trees (FAST): A splitting procedure to grow a binary tree using a suitable mathematical property of the impurity proportional reduction measure to find out the optimal split at each node without trying out necessarily all the candidate splits.

Production Rule: A tree path characterized by a sequence of interactions between attributes yielding to a specific class/value of the response.

Pruning: A top-down or bottom-up selective algorithm to reduce the dimensionality of a tree structure in terms of the number of its terminal nodes.

Random Forest: An ensemble of unpruned trees obtained by randomly resampling training instances and attributes. The overall performance of the method derives from averaging the generalization errors obtained in each run.

Recursive Partitioning Algorithm: A recursive algorithm to form disjoint and exhaustive subgroups of objects from a given group in order to build up a decision tree.

Regression Tree: An oriented tree structure obtained by recursively partitioning a sample of objects on the basis of a sequential partitioning of the attributes space such to obtain internally homogenous groups and externally heterogeneous groups of objects with respect to a numerical attribute.

Deep Web Mining through Web Services

Monica Maceli
Drexel University, USA

Min Song
New Jersey Institute of Technology & Temple University, USA

INTRODUCTION

With the increase in Web-based databases and dynamically-generated Web pages, the concept of the "deep Web" has arisen. The deep Web refers to Web content that, while it may be freely and publicly accessible, is stored, queried, and retrieved through a database and one or more search interfaces, rendering the Web content largely hidden from conventional search and spidering techniques. These methods are adapted to a more static model of the "surface Web", or series of static, linked Web pages. The amount of deep Web data is truly staggering; a July 2000 study claimed 550 billion documents (Bergman, 2000), while a September 2004 study estimated 450,000 deep Web databases (Chang, He, Li, Patel, & Zhang, 2004).

In pursuit of a truly searchable Web, it comes as no surprise that the deep Web is an important and increasingly studied area of research in the field of Web mining. The challenges include issues such as new crawling and Web mining techniques, query translation across multiple target databases, and the integration and discovery of often quite disparate interfaces and database structures (He, Chang, & Han, 2004; He, Zhang, & Chang, 2004; Liddle, Yau, & Embley, 2002; Zhang, He, & Chang, 2004).

Similarly, as the Web platform continues to evolve to support applications more complex than the simple transfer of HTML documents over HTTP, there is a strong need for the interoperability of applications and data across a variety of platforms. From the client perspective, there is the need to encapsulate these interactions out of view of the end user (Balke & Wagner, 2004). Web services provide a robust, scalable and increasingly commonplace solution to these needs.

As identified in earlier research efforts, due to the inherent nature of the deep Web, dynamic and ad hoc information retrieval becomes a requirement for mining such sources (Chang, He, & Zhang, 2004; Chang, He, Li, Patel, & Zhang, 2004). The platform and program-agnostic nature of Web services, combined with the power and simplicity of HTTP transport, makes Web services an ideal technique for application to the field of deep Web mining. We have identified, and will explore, specific areas in which Web services can offer solutions in the realm of deep Web mining, particularly when serving the need for dynamic, ad-hoc information gathering.

BACKGROUND

Web Services

In the distributed computing environment of the internet, Web services provide for application-to-application interaction through a set of standards and protocols that are agnostic to vendor, platform and language (W3C, 2004). First developed in the late 1990s (with the first version of SOAP being submitted to the W3C in 2000), Web services are an XML-based framework for passing messages between software applications (Haas, 2003). Web services operate on a request/response paradigm, with messages being transmitted back and forth between applications using the common standards and protocols of HTTP, eXtensible Markup Language (XML), Simple Object Access Protocol (SOAP), and Web Services Description Language (WSDL) (W3C, 2004). Web services are currently used in many contexts, with a common function being to facilitate inter-application communication between the large number of vendors, customers, and partners that interact with today's complex organizations (Nandigam, Gudivada, & Kalavala, 2005). A simple Web service is illustrated by the below diagram; a Web service provider (consisting of a Web server connecting to a database server) exposes an XML-based API (Application Programming Interface)

to a catalog application. The application manipulates the data (in this example, results of a query on a collection of books) to serve both the needs of an end user, and those of other applications.

Semantic Web Vision

Web services are considered an integral part of the semantic Web vision, which consists of Web content described through markup in order to become fully machine-interpretable and processable. The existing Web is endowed with only minimal semantics; the semantic Web movement endeavors to enhance and increase this semantic data. Additionally, the semantic Web will provide great strides in overcoming the issues of language complexity and ambiguity that currently inhibit effective machine processing. The projected result will be a more useful Web where information can be intelligently shared, combined, and identified.

Semantic Web services are Web services combined with ontologies (high-level metadata) that describe Web service content, capabilities and properties, effectively merging the technical strengths of Web services with the descriptive capacity of ontologies (Narayanan & McIlraith, 2002; Terziyan & Kononenko, 2003). Ontologies are vital to the concept of the semantic Web, describing and defining the relationships and concepts which allow for interoperability. Not surprisingly, there has been a great deal of recent research exploring topics such as how to construct, model, use, personalize, maintain, classify, and transform the semantic Web ontologies (Acuña & Marcos, 2006; Alani, 2006; Jiang & Tan, 2006; Lei, 2005; Pathak & Koul, 2005). Ultimately, by describing both what the Web service provides and how to interact with it in a machine-readable fashion, semantic Web services will allow automation in such areas as Web service discovery, integration and interoperation.

Deep Web Challenges

As identified in earlier works (Chang, He, & Zhang, 2004; Chang, He, Li, Patel, & Zhang, 2004), dynamic and ad-hoc integration will be a requirement for large-scale efforts to mine the deep Web. Current deep Web mining research is exploring Web query interface integration, which allows for the querying of multiple hidden Web databases through a unified interface (Bergholz & Chidlovskii, 2004; He, Chang, & Han, 2004; Liu & Chen-Chuan-Chang, 2004). The below figure illustrates the current challenges of the distributed Web environment. The Web consists of both static Web pages and database-specific interfaces; while static pages are easily mined, the Web databases are hidden behind one or more dynamic querying interfaces, presenting an obstacle to mining efforts.

These studies exploit such techniques as interface extraction, schema matching, and interface unification to integrate the variations in database schema and display into a meaningful and useful service for the user (He, Zhang, & Chang, 2004). Interface extraction seeks to automatically extract the relevant attributes from the HTML of the query interface Web page, schema matching identifies semantic similarity between these attributes, and interface unification refers to the construction of a single unified interface based upon the identified matches (He, Zhang, & Chang, 2004). Such unified interfaces then serve as a mediator between the user and the multiple deep Web databases that are being queried; the request and aggregation of this data

Figure 1. High-level Web service example

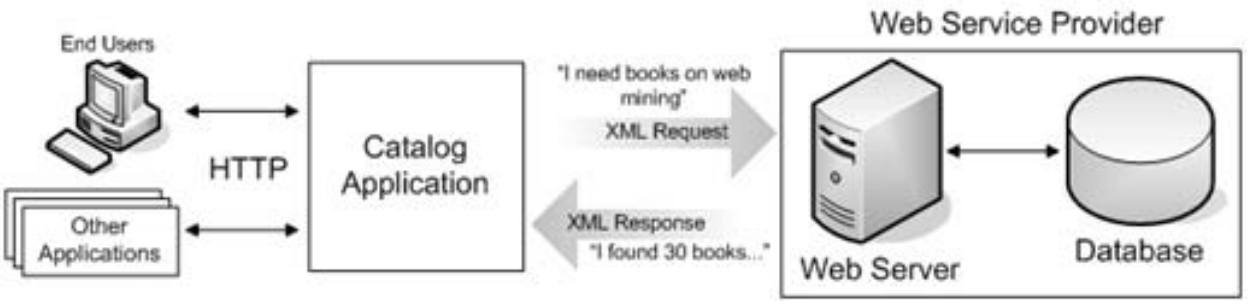
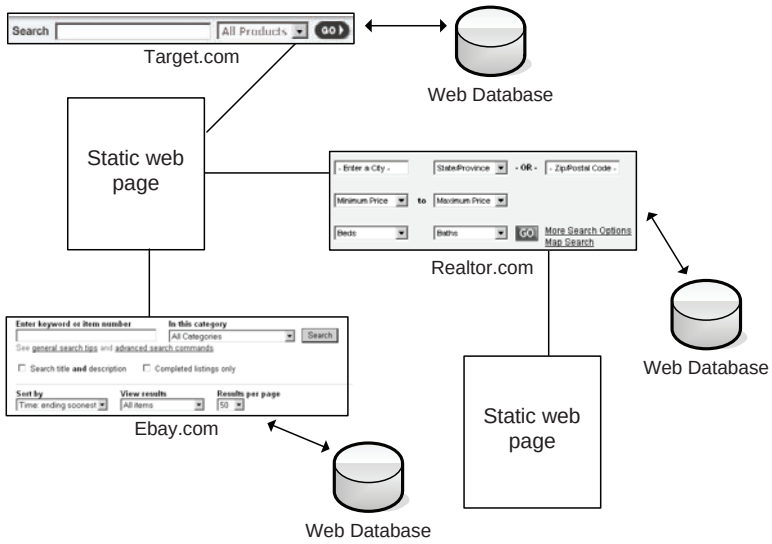


Figure 2. Web databases and associated interfaces of the deep Web, as well as static Web pages of the surface Web



is thusly hidden from the user, providing a seamless and usable interface. On a large scale, however, such solutions provide many challenges in the huge and data-heavy environment of the Web. In particular, schema matching across disparate databases can be challenging and require techniques such as statistical analysis of patterns and exploration of context information to be successful.

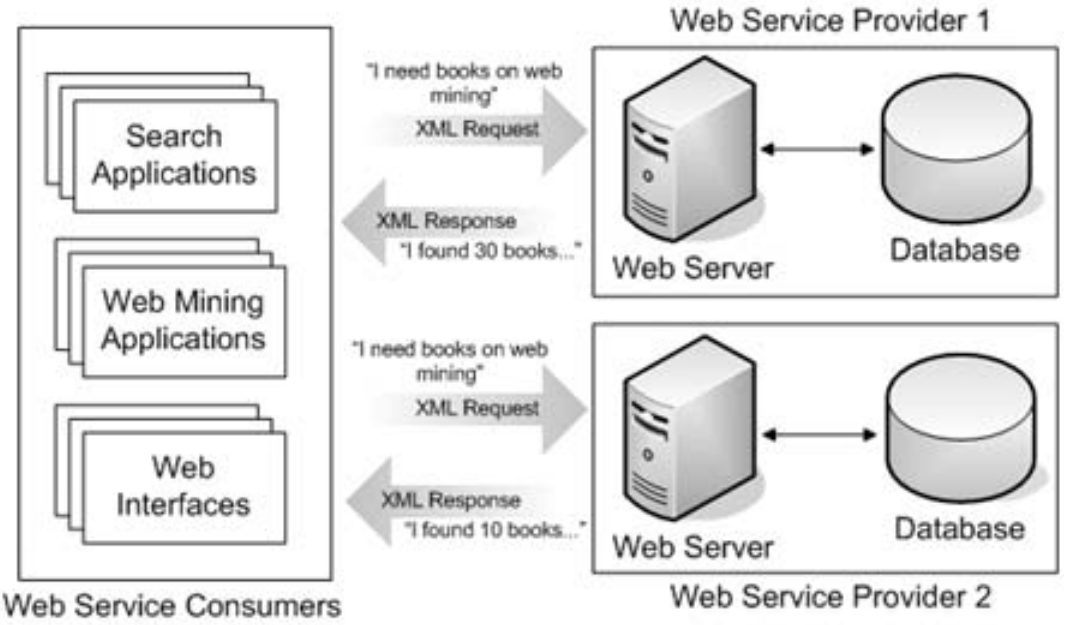
MAIN FOCUS

Current deep Web mining research has found two primary requirements for success - the ability to retrieve data in a dynamic fashion (as Web sources are continually evolving) and in an ad-hoc manner (as queries are tailored to users' varying needs) (Chang, He, & Zhang, 2004). By providing standardized APIs (Application Programming Interfaces) against which to dynamically request data, Web services can make significant contributions to the field of deep Web mining. Web services can encapsulate database-specific algorithms and structure, providing only a standardized search API to a Web mining tool or search interface. Semantic Web services are an effective means of exposing the structured database information of the deep Web to both search engines and dynamic search applications (and, as is the nature of Web services, to any other desired application).

The current research in deep Web mining endeavors to discover semantics and matches between the varying query interfaces of the deep Web, in order to both identify appropriate sources and to translate the user's query across schemas effectively and logically (Bergholz & Chidlovskii, 2004; He, Chang, & Han, 2004; He, Zhang, & Chang, 2004). Web services can be considered an effective data delivery system in such an environment. The below diagram illustrates a high-level view of a system in which semantically-endowed Web services provide dynamic results to a host of applications, including deep Web mining applications.

As discussed earlier, Web services offer a myriad of benefits and their success lies in their ability to provide dynamic and extremely flexible interaction between distributed and heterogeneous Web systems. Interoperability issues can be largely negated when software components are exposed as Web services (Nandigam, Gudivada, & Kalavala, 2005). Web services have the ability to take advantage of pre-existing systems by providing a standardized wrapper of communication on top of existing applications. In this manner they do not require the replacement or rewriting of systems, and subsequently take a relatively small amount of time to implement. The platform-independent nature of Web services allows for a true separation of display and data;

Figure 3. Web service APIs providing data to multiple applications



the result is that the same Web service can be called from multiple applications, platforms, and device types, and combined and used in a truly unlimited number of ways. In addition to deep Web mining, Web services have application in the field of meta-searching and federated search, which are analogous library solutions for integration of disparate databases.

Additionally, there is similar research in the area of Web personalization and intelligent search, where the appropriate Web services will be chosen at run-time based on user preferences (Balke &Wagner, 2004). These techniques allow multiple databases and data stores to be queried through one, seamless interface that is tailored to a particular search term or area of interest. In addition to being a more efficient method of searching the Web, such interfaces more closely match users' current mental model and expectations of how a search tool "should" behave (Nielsen, 2005).

Web Service Challenges

As we will explore in the future trends section of the study, there are significant efforts underway in the standardization and proliferation of Web services, and these factors remain the chief obstacles to such a system. Although Web services are quite popular in achieving the interoperability of organizations' internal applications, such services are not as consistently provided publicly. There are an increasing number of available Web services, but they are not provided by every system or database-driven application of the deep Web (Nandigam, Gudivada, & Kalavala, 2005). Alternative techniques such as mediators and screen-scrapers exist to submit deep Web queries, but are much more inefficient and labor-intensive.

When dealing with the many databases of the deep Web, the query often must be translated to be meaningful to every target system, which can require significant processing (Chang, He, & Zhang, 2004). Not surprisingly, this translation can also have an effect on the quality and relevance of the results provided. Similarly, the results returned are only as good as those of the target system and the de-duplication of data is a cost-heavy requirement as well.

FUTURE TRENDS

Previous studies have shown that the data-rich, structured environment of the deep Web continues to grow

at a rate faster than that of the surface Web, indicating a wealth of opportunities for exploring deep Web mining (Bergman, 2000; Chang, He, Li, Patel, & Zhang, 2004). Significant progress remains to be made in the realm of Web services development and standardization as well. Both industry and academic research initiatives have explored methods of standardizing Web service discovery, invocation, composition, and monitoring, with the World Wide Web Consortium (W3C) leading such efforts (W3C, 2004). The end goal of these endeavors are Web-based applications that can automatically discover an appropriate Web service, obtain the proper steps to invoke the service, create composite interoperability of multiple services when necessary, and monitor the service properties during the lifespan of the interaction (Alesso, 2004; Hull & Su, 2005).

Along with standardization initiatives and frontier research, there is a general increase in the popularity of Web services, both in industry and academia, with a variety of public services currently available (Fan & Kambhampati, 2005). Businesses and organizations continue to open entry to their capabilities through Web services; a well-used example is Amazon.com's Web service which provides access to its product database, and claimed 65,000 developers as of November 2004 (Nandigam, Gudivada, & Kalavala, 2005). Research on the current state of Web services has focused on the discovery of such existing services, with UDDI (Universal Description, Discovery, and Integration) providing a registry for businesses to list their services, aiding in discovery and interaction of systems (Nandigam, Gudivada, & Kalavala, 2005).

All of these developments will provide a strong framework for dynamic mining of the deep Web through Web services, and can be summarized as follows:

1. Increase in the number of publicly available Web services, as well as those used for organizations' internal applications
2. Higher percentage of quality Web services registered and discoverable through UDDI
3. Further standardization of semantic Web services' ontologies, ultimately allowing for machines to select, aggregate, and manipulate Web service data in a dynamic fashion, including deep Web information.
4. Further improvements to intelligent search, where users' queries can be tailored and distributed to multiple, relevant sources at run-time.
5. Continuing research in exposing and manipulating deep Web data as well as applying semantics to the variety of information types and sources currently available on the Web

CONCLUSION

In the distributed and deep data environment of the internet, there is a strong need for the merging and collation of multiple data sources into a single search interface or request. These data stores and databases are often part of the "deep Web" and are out of the range of classic search engines and crawlers. In the absence of authoritative deep Web mining tools, Web services and other techniques can be leveraged to provide dynamic, integrated solutions and the interoperability of multiple systems. Semantic Web services can expose the structured, high-quality data of the deep Web to Web mining efforts. These systems can provide simultaneous searching of multiple databases and data stores. However, such solutions are not without their own challenges and limitations, which must be carefully managed and addressed, and significant standardization efforts are still required.

REFERENCES

Acuña, C. J. and Marcos, E. (2006). Modeling semantic web services: a case study. In Proceedings of the 6th international Conference on Web Engineering (Palo Alto, California, USA, July 11 - 14, 2006). ICWE '06. ACM Press, New York, NY, 32-39.

Alani, H. (2006). Position paper: ontology construction from online ontologies. In Proceedings of the 15th international Conference on World Wide Web (Edinburgh, Scotland, May 23 - 26, 2006). WWW '06. ACM Press, New York, NY, 491-495.

Alesso, H. Peter (2004). *Preparing for Semantic Web Services*. Retrieved March 1, 2006 from http://www.sitepoint.com/article/semantic-Web-services.

Balke, W. & Wagner, M. (2004). Through different eyes: assessing multiple conceptual views for querying

Web services. In *Proceedings of the 13th international World Wide Web Conference on Alternate Track Papers &Amp; Posters* (New York, NY, USA, May 19 - 21, 2004). WWW Alt. '04. ACM Press, New York, NY, 196-205.

Bergholz, A. & Chidlovskii, B. (2004). Learning query languages of Web interfaces. In *Proceedings of the 2004 ACM Symposium on Applied Computing* (Nicosia, Cyprus, March 14 - 17, 2004). SAC '04. ACM Press, New York, NY, 1114-1121.

Bergman, M. K. (2000). *The Deep Web: Surfacing Hidden Value*. Technical report, BrightPlanet LLC.

Chang, K. C., He, B., & Zhang, Z. (2004). Mining semantics for large scale integration on the Web: evidences, insights, and challenges. *SIGKDD Explor. Newsl.* 6, 2 (Dec. 2004), 67-76.

Chang, K. C., He, B., Li, C., Patel, M., & Zhang, Z. (2004). Structured databases on the Web: observations and implications. *SIGMOD Rec.* 33, 3 (Sep. 2004), 61-70.

Fan, J. and Kambhampati, S. (2005). A snapshot of public Web services. *SIGMOD Rec.* 34, 1 (Mar. 2005), 24-32.

Haas, Hugo. (2003). Web Services: setting and re-setting expectations. http://www.w3.org/2003/Talks/0618-hh-wschk/

He, B., Chang, K. C., & Han, J. (2004). Mining complex matchings across Web query interfaces. In *Proceedings of the 9th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery* (Paris, France). DMKD '04. ACM Press, New York, NY, 3-10.

He, B., Zhang, Z., & Chang, K. C. (2004). Knocking the door to the deep Web: integrating Web query interfaces. In *Proceedings of the 2004 ACM SIGMOD international Conference on Management of Data* (Paris, France, June 13 - 18, 2004). SIGMOD '04. ACM Press, New York, NY, 913-914.

Hull, R. & Su, J. (2005). Tools for composite Web services: a short overview. *SIGMOD Rec.* 34, 2 (Jun. 2005), 86-95.

Jiang, X. and Tan, A. (2006). Learning and inferencing in user ontology for personalized semantic web services. In Proceedings of the 15th international Conference on World Wide Web (Edinburgh, Scotland, May 23 - 26, 2006). WWW '06. ACM Press, New York, NY, 1067-1068.

Lei, Y. (2005). An instance mapping ontology for the semantic web. In Proceedings of the 3rd international Conference on Knowledge Capture (Banff, Alberta, Canada, October 02 - 05, 2005). K-CAP '05. ACM Press, New York, NY, 67-74.

Liddle, W. S., Yau, S. H., & Embley, W. D. (2002). On the Automatic Extraction of Data from the Hidden Web, Lecture Notes in Computer Science, Volume 2465, Jan 2002, Pages 212 - 226

Liu, B. & Chen-Chuan-Chang, K. (2004). Editorial: special issue on Web content mining. *SIGKDD Explor. Newsl.* 6, 2 (Dec. 2004), 1-4.

Nandigam, J., Gudivada, V. N., & Kalavala, M. (2005). Semantic Web services. *J. Comput. Small Coll.* 21, 1 (Oct. 2005), 50-63.

Narayanan, S. & McIlraith, S. A. 2002. Simulation, verification and automated composition of Web services. In *Proceedings of the 11th international Conference on World Wide Web* (Honolulu, Hawaii, USA, May 07 - 11, 2002). WWW '02. ACM Press, New York, NY, 77-88.

Nielsen, Jakob. (2005). *Mental Models for Search are Getting Firmer*. Retrieved February 21, 2006 from http://www.useit.com/alertbox/20050509.html.

Pathak, J., Koul, N., Caragea, D., and Honavar, V. G. (2005). A framework for semantic web services discovery. In Proceedings of the 7th Annual ACM international Workshop on Web information and Data Management (Bremen, Germany, November 04 - 04, 2005). WIDM '05. ACM Press, New York, NY, 45-50.

Terziyan, V., & Kononenko, O. (2003). Semantic Web Enabled Web Services: State-of-Art and Industrial Challenges, Lecture Notes in Computer Science, Volume 2853, Jan 2003, Pages 183 - 197

W3C. (2004). *Web Services Architecture, W3C Working Group Note 11 February 2004*. Retrieved February 15, 2005 from http://www.w3.org/TR/ws-arch/.

Zhang, Z., He, B., & Chang, K. C. (2004). Understanding Web query interfaces: best-effort parsing with hidden syntax. In *Proceedings of the 2004 ACM SIGMOD international Conference on Management*

of Data (Paris, France, June 13 - 18, 2004). SIGMOD '04. ACM Press, New York, NY, 107-118.

KEY TERMS

Deep Web: Deep Web is content that resides in searchable databases, the results from which can only be discovered by a direct query.

Ontologies: Ontologies are meta-data which provide a machine-processable and controlled vocabulary of terms and semantics; they are critical to the semantic Web vision as ontologies support both human and computer understanding of data.

Semantic Web: Semantic Web is a self describing machine processable Web of extensible dynamic information; a huge logic based database of semantically marked up information, ready for output and processing.

Surface Web: The surface Web refers to the static Web pages of the Internet which are linked together by hyperlinks and are easily crawled by conventional Web crawlers.

UDDI (Universal Description, Discovery, and Integration): UDDI, or Universal Description, Discovery, and Integration, is an XML-based registry for Web services, providing a means for finding and using public Web services.

Web Mining: Web mining is the automated discovery of useful information from the Web documents using data mining and natural language processing techniques.

Web Services: Web services can be defined as a standardized way of integrating Web-based applications using the XML, SOAP, WSDL and UDDI open standards over an Internet protocol backbone.

DFM as a Conceptual Model for Data Warehouse

Matteo Golfarelli
University of Bologna, Italy

INTRODUCTION

Conceptual modeling is widely recognized to be the necessary foundation for building a database that is well-documented and fully satisfies the user requirements. In particular, from the designer point of view the availability of a conceptual model provides a higher level of abstraction in describing the warehousing process and its architecture in all its aspects.

Typically conceptual models rely on a graphical notation that facilitates writing, understanding, and managing conceptual schemata by both designers and users. The *Entity/Relationship* (E/R) model (Chen, 1976) is widespread in the enterprises as a conceptual formalism to provide standard documentation for relational information systems; nevertheless, as E/R is oriented to support queries that navigate associations between data rather than synthesize them, it is not well-suited for data warehousing (Kimball, 1998). Actually, the E/R model has enough expressivity to represent most concepts necessary for modeling a *Data Warehouse* (DW); on the other hand, in its basic form, it is not able to properly emphasize the key aspects of the multidimensional model, so that its usage for DWs is expensive from the point of view of the graphical notation and not intuitive (Rizzi, 2006).

Some designers claim that star schemata are expressive enough for conceptual modeling. Actually, a star schema is just a (denormalized) relational schema, so it merely defines a set of relations and integrity constraints. Using star schema for conceptual modeling is like starting to build a complex software by writing the code, without the support of any static, functional, or dynamic model, which typically leads to very poor results from the points of view of adherence to user requirements, maintenance, and reuse. For all these reasons, in the last few years the research literature has proposed several original approaches for modeling a DW, some based on extensions of known conceptual formalisms (e.g. E/R, *Unified Modeling Language* (UML)), some based on ad hoc ones. Remarkably, a comparison of the different models made by Abello (2006) pointed out that, abstracting from their graphical form, the core expressivity is similar, thus proving that the academic community reached an informal agreement on the required expressivity.

This paper discusses the expressivity of an ad hoc conceptual model, the Dimensional Fact Model (DFM), in order to let the user verify the usefulness of a conceptual modeling step in DW design. After a brief listing of the main conceptual model proposals, the basic and advanced features in DW conceptual modeling are introduced and described by examples. Finally, the current trends in DW conceptual modeling are reported and the conclusions are drawn.

BACKGROUND

In the last few years multidimensional modeling attracted the attention of several researchers that defined different solutions each focusing on the set of information they considered strictly relevant. Some of these solutions have no (Agrawal, 1997; Pedersen, 1999) or limited (Cabibbo, 1998) graphical support, and are aimed at establishing a formal foundation for representing cubes and hierarchies and an algebra for querying them. On the other hand, we believe that a distinguishing feature of conceptual models is that of providing a graphical support to be easily understood by both designers and users when discussing and validating requirements. So we will classify "strict" conceptual models for DWs according to the graphical formalism they rely on that could be either E/R, object-oriented or ad hoc. Some claim that E/R extensions should be adopted since (1) E/R has been tested for years; (2) designers are familiar with E/R; (3) E/R has proved to be flexible and powerful enough to adapt to a variety of application domains; and (4) several important research results were obtained for the E/R model (Franconi, 2004; Sapia, 1999; Tryfona, 1999). On the other hand, advocates of object-oriented models argue that (1) they are more

Table 1. Comparison between conceptual models (√: supported feature; −: not supported or not explained how to support; p partially supported; QL query language; A: algebra; C: calculus)

	Agrawal 1997	Cabibbo 1998	Pedersen 1999	Franconi 2004	Sapia 1999	Tryfona 1999	Luján 2006	Abello 2006	Tsois 2001	Hüsemann 2000	Rizzi 2006
Reference conceptual model	–	Maths	Maths	E/R	E/R	E/R	UML	UML	–	–	–
Formalism adopted to define operations over data	A	A C	A	–	–	–	–	A	–	–	QL
Shows how data can be aggregated	P	√	√	√	√	√	√	√	√	√	√
Allows multiple aggregation paths along dimensions	√	√	√	√	√	√	√	√	√	√	√
Describes hierarchies besides aggregation issues	–	√	–	–	√	√	–	√	√	√	√
Allows multiple measure to be represented	P	–	√	√	√	√	√	√	7	√	√
Models summarizzability semantics	–	–	–	–	–	P	√	√	–	√	√
Treats symmetrically measures and dimensions	√	–	√	–	–	P	–	P	–	–	–
Multiple subjects (fact) can be represented	–	√	–	–	√	–	√	√	–	√	√
Models many-to-many rel. within hierarchies	P	–	√	√	–	√	√	√	√	–	√

expressive and better represent static and dynamic properties of information systems; (2) they provide powerful mechanisms for expressing requirements and constraints; (3) object-orientation is currently the dominant trend in data modeling; and (4) UML, in particular, is a standard and is naturally extensible (Luján-Mora, 2006; Abello, 2006). Finally, we believe that ad hoc models compensate for designers' lack of familiarity since (1) they achieve better notational economy; (2) they give proper emphasis to the peculiarities of the multidimensional model, and thus (3) they are more intuitive and readable by non-expert users (Rizzi, 2006; Hüsemann, 2000; Tsois, 2001).

Table 1 reports a comparison of the conceptual models discussed according to a subset of the criteria[1] proposed by Abello (2006). The subset has been defined considering the most common features (i.e. those supported by most of the models) that define the core expressivity for multidimensional conceptual modeling.

MAIN FOCUS

The Basics of the Dimensional Fact Model

The Dimensional Fact Model is a graphical conceptual model, specifically devised for multidimensional design, aimed at:

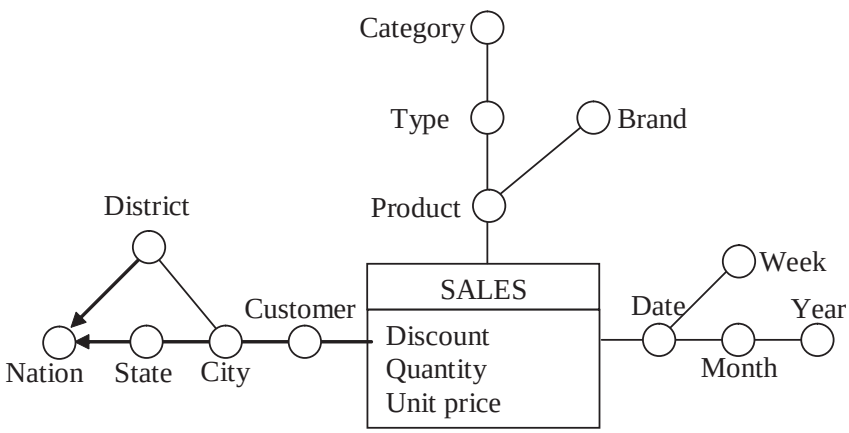
- effectively supporting conceptual design;
- providing an environment in which user queries can be intuitively expressed;
- supporting the dialogue between the designer and the end-users to refine the specification of requirements;
- creating a stable platform to ground logical design;
- providing an expressive and non-ambiguous design documentation.

DFM was first proposed in 1998 by Golfarelli and Rizzi and continuously enriched and refined during the following years in order to optimally suit the variety of modeling situations that may be encountered in real projects of small to large complexity.

The representation of reality built using the DFM consists of a set of fact schemata. The basic concepts modeled are facts, measures, dimensions, and hierarchies. In the following we intuitively define these concepts, referring the reader to Figure 1 that depicts a simple fact schema for modeling invoices at line granularity; a formal definition of the same concepts can be found in (Rizzi, 2006).

Definition 1. A *fact* is a focus of interest for the decision-making process; typically, it models a set of events occurring in the enterprise world. A fact is graphically represented by a box with two sections, one for the fact name and one for the measures.

Figure 1. A basic fact schema for the SALES fact



Examples of facts in the trade domain are sales, shipments, purchases; in the financial domain: stock exchange transactions, contracts for insurance policies. It is essential for a fact to have some dynamic aspects, i.e., to evolve somehow across time.

The concepts represented in the data source by frequently-updated archives are good candidates for facts; those represented by almost-static archives are not. As a matter of fact, very few things are completely static; even the relationship between cities and regions might change, if some borders were revised. Thus, the choice of facts should be based either on the average periodicity of changes, or on the specific interests of analysis.

Definition 2. A *measure* is a numerical property of a fact, and describes one of its quantitative aspects of interests for analysis. Measures are included in the bottom section of the fact.

For instance, each invoice line is measured by the number of units sold, the price per unit, the net amount, etc. The reason why measures should be numerical is that they are used for computations. A fact may also have no measures, if the only interesting thing to be recorded is the occurrence of events; in this case the fact scheme is said to be empty and is typically queried to count the events that occurred.

Definition 3. A *dimension* is a fact property with a finite domain, and describes one of its analysis coordinates. The set of dimensions of a fact determine its finest representation granularity. Graphically, dimensions are represented as circles attached to the fact by straight lines.

Typical dimensions for the invoice fact are product, customer, agent. Usually one of the dimensions of the fact represents the time (at any granularity) that is necessary to extract time series from the DW data.

The relationship between measures and dimensions is expressed, at the instance level, by the concept of event.

Definition 4. A *primary event* is an occurrence of a fact, and is identified by a tuple of values, one for each dimension. Each primary event is described by one value for each measure.

Primary events are the elemental information which can be represented (in the cube metaphor, they correspond to the cube cells). In the invoice example they model the invoicing of one product to one customer made by one agent on one day.

Aggregation is the basic OLAP operation, since it allows significant information to be summarized from large amounts of data. From a conceptual point of view, aggregation is carried out on primary events thanks to the definition of dimension attributes and hierarchies.

Definition 5. A *dimension attribute* is a property, with a finite domain, of a dimension. Like dimensions, it is represented by a circle.

For instance, a product is described by its type, category, and brand; a customer, by its city and its nation. The relationships between dimension attributes are expressed by hierarchies.

Definition 6. A *hierarchy* is a directed graph, rooted in a dimension, whose nodes are all the dimension attributes that describe that dimension, and whose arcs model many-to-one associations between pairs of dimension attributes. Arcs are graphically represented by straight lines.

Hierarchies should reproduce the pattern of inter-attribute functional dependencies expressed by the data source. Hierarchies determine how primary events can be aggregated into secondary events and selected significantly for the decision-making process.

Definition 7. Given a set of dimension attributes, each tuple of their values identifies a *secondary event* that aggregates all the corresponding primary events. Each secondary event is described by a value for each measure, that summarizes the values taken by the same measure in the corresponding primary events.

The dimension in which a hierarchy is rooted defines its finest aggregation granularity, while the other dimension attributes progressively define coarser ones. For instance, thanks to the existence of a many-to-one association between products and their categories, the invoicing events may be grouped according to the

category of the products. When two nodes a_1, a_2 of a hierarchy share the same descendent a_3 (i.e. when two dimension attributes within a hierarchy are connected by two or more alternative paths of many-to-one associations) this is the case of a convergence, meaning that for each instance of the hierarchy we can have different values for a_1, a_2, but we will have only one value for a_3. For example, in the geographic hierarchy on dimension customer (Figure 1): customers live in cities, which are grouped into states belonging to nations. Suppose that customers are also grouped into sales districts, and that no inclusion relationships exist between districts and cities/states; on the other hand, sales districts never cross the nation boundaries. In this case, each customer belongs to exactly one nation whichever of the two paths is followed (customer→city→state→nation or customer→sale district→nation).

It should be noted that the existence of apparently equal attributes does not always determine a convergence. If in the invoice fact we had a brand city attribute on the product hierarchy, representing the city where a brand is manufactured, there would be no convergence with attribute (customer) city, since a product manufactured in a city can obviously be sold to customers of other cities as well.

Advanced Features of the Dimensional Fact Model

In this section we introduce, with the support of Figure 2, some additional constructs of DFM that enable us to best capture those peculiarities of the application domain and that turn out to be necessary for the designer during logical design.

In several cases it is useful to represent additional information about a dimension attribute, though it is not interesting to use such information for aggregation. For instance, the user may ask to know the address of each store, but will hardly be interested in aggregating sales according to the address of the store.

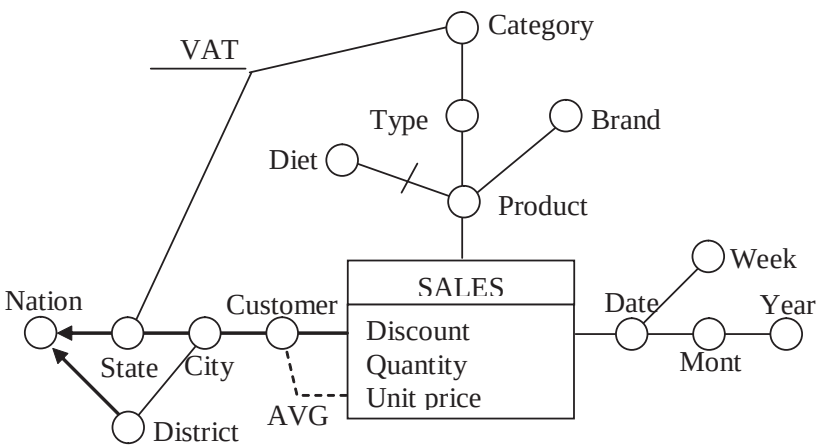
Definition 8. A *descriptive attribute* specifies a property of a dimension attribute, to which it is related by an one-to-one association. Descriptive attributes are not used for aggregation; they are always leaves of their hierarchy and are graphically represented by horizontal lines.

Usually a descriptive attribute either has a continuously-valued domain (for instance, the weight of a product), or is related to a dimension attribute by a one-to-one association (for instance, the address of a customer).

Definition 9. A *cross-dimension attribute* is an (either dimensional or descriptive) attribute whose value is determined by the combination of two or more dimension attributes, possibly belonging to different hierarchies. It is denoted by connecting the arcs that determine it by a curved line.

For instance, if the VAT on a product depends on both the product category and the state where the prod-

Figure 2. The complete fact schema for the SALES fact



uct is sold, it can be represented by a cross-dimension attribute.

Definition 10. An *optional arc* models the fact that an association represented within the fact scheme is undefined for a subset of the events. An optional arc is graphically denoted by marking it with a dash.

For instance, attribute diet takes a value only for food products; for other products, it is undefined.

In most cases, as already said, hierarchies include attributes related by many-to-one associations. On the other hand, in some situations it is necessary to include also those attributes that, for a single value taken by their father attribute, take several values.

Definition 11. A *multiple arc* is an arc, within a hierarchy, modeling a many-to-many association between the two dimension attributes it connects. Graphically, it is denoted by doubling the line that represents the arc.

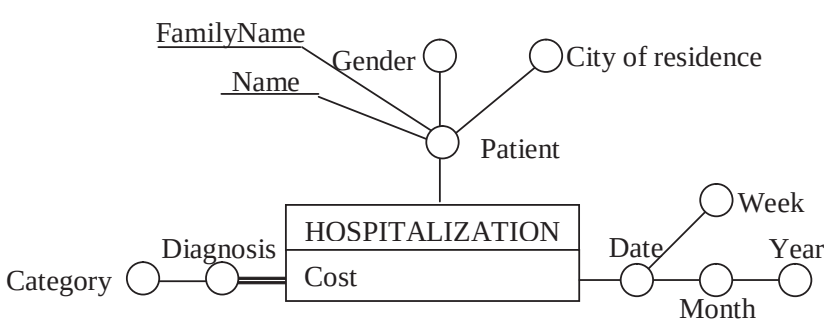
Consider the fact schema modeling the hospitalization of patient in a hospital, represented in Figure 3, whose dimensions are date, patient and diagnosis. Users will probably be interested in analyzing the cost of staying in hospital for different diagnoses, but, since for each hospitalization the doctor in charge of the patient could specify up to 7 diagnoses, the relationship between the fact and the diagnoses must be modeled as a multiple arc.

Summarizability is the property of correcting summarizing measures along hierarchies (Lenz, 1997). Aggregation requires defining a proper operator to compose the measure values characterizing primary events into measure values characterizing each secondary event. Usually, most measures in a fact scheme are *additive*, meaning that they can be aggregated on different dimensions using the SUM operator. An example of additive measure in the sale scheme is *quantity*: the quantity sold in a given month is the sum of the quantities sold on all the days of that month. A measure may be *non-additive* on one or more dimensions. Examples of this are all the measures expressing a level, such as an inventory level, a temperature, etc. An inventory level is non-additive on time, but it is additive on the other dimensions. A temperature measure is non-additive on all the dimensions, since adding up two temperatures hardly makes sense. However, this kind of non-additive measures can still be aggregated by using operators such as average, maximum, minimum. The set of operators applicable to a given measure can be directly represented on the fact scheme by labeling a dashed line connecting a measure and a dimension with the name of the operators (see Figure 2). In order to increase readability the information is omitted for the SUM operator that is chosen as default. If the schema readability is affected in any way we recommend substituting the graphical representation with a tabular one showing measures on the rows and dimensions on the columns: each cell will contain the set of allowed operators for a given dimension and a given measure.

FUTURE TRENDS

Future trends in DWing were recently discussed by the academic community in the Perspective Seminar "Data Warehousing at the Crossroads" that took place

Figure 3. The fact schema for the HOSPITALIZATION fact



at Dagstuhl, Germany, in August 2004. Here we report the requirements related to conceptual modeling expressed by the group:

- Standardization. Though several conceptual models have been proposed, none of them has been accepted as a standard. On the other hand, the ''Common Warehouse Metamodel'' (OMG, 2001) standard is too general, and not conceived as a conceptual model (Abello, 2006). An effort is needed to promote the adoption of a design methodology centered on conceptual modeling.
- Implementation of CASE tools. No commercial case tools currently support conceptual models; on the other hand, a tool capable of drawing conceptual schemata and deploying relational tables directly from them has already been prototyped and would be a valuable support for both the research and industrial communities.
- Conceptual models for the DW process. Conceptual modeling can support all the steps in the DW process. For example design and implementation of ETL procedures could be simplified adopting a proper conceptual modeling phase. Though some approaches are already available (Trujillo, 2003; Simitsis, 2007), the topic requires further investigation. Similarly, the definition of security polices, that are particularly relevant in DW systems, can be modeled at the conceptual level (Fernandez, 2007). Finally, ad-hoc conceptual models may be applied in the area of analytic applications built on top of DW. For example, the design of What-if analysis applications is centered on the definition of the simulation model (Golfarelli, 2006), whose definition requires an ad-hoc expressivity that is not available in any known conceptual models.

CONCLUSION

In this paper we have proposed the DFM as a model that covers the core expressivity required for supporting the conceptual design of DW applications and that presents further constructs for effectively describing the designer solutions.

Since 1998, the DFM has been successfully adopted in several different DW projects in the fields of large-scale retail trade, telecommunications, health, justice, and education, where it has proved expressive enough to capture a wide variety of modeling situations. Remarkably, in most projects the DFM was also used to directly support dialogue with end-users aimed at validating requirements, and to express the expected workload for the DW to be used for logical and physical design. Overall, our on-the-field experience confirmed that adopting conceptual modeling within a DW project brings great advantages since:

- Conceptual schemata are the best support for discussing, verifying, and refining user specifications since they achieve the optimal trade-off between expressivity and clarity. Star schemata could hardly be used to this purpose.
- Conceptual schemata are an irreplaceable component of the project documentation.
- They provide a solid and platform-independent foundation for logical and physical design.
- They make turn-over of designers and administrators on a DW project quicker and simpler.

REFERENCES

Agrawal, R., Gupta, A., & Sarawagi, S. (1997). Modeling multidimensional databases. In *Proceedings of the 13th International Conference on Data Engineering*, (pp. 232-243), Birmingham U.K.

Abello, A., Samos J., & Saltor F. (2006). YAM2: A multidimensional conceptual model extending UML. *Information System*, 31(6), 541-567.

Cabibbo, L., & Torlone, R. (1998). A logical approach to multidimensional databases. In *Proceedings International Conference on Extending Database Technology* (pp. 183-197). Valencia, Spain.

Chen, P. (1976). The entity-relationship model – towards a unified view of data. *ACM Transactions on Database Systems (TODS)*, 1(1), 9-36.

Fernandez-Medina, E., Trujillo J., Villarroel R., & Piattini M. (2007). Developing secure data warehouses with a UML extension. *Information System*, 32(6), 826-856.

Franconi, E., & Kamble, A. (2004). A data warehouse conceptual data model. In *Proceedings International Conference on Statistical and Scientific Database Management* (pp. 435-436). Santorini Island, Greece.

Golfarelli, M., Rizzi S., & Proli A (2006). Designing What-if Analysis: Towards a Methodology. In *Proceedings 9th International Workshop on Data Warehousing and OLAP*, (pp. 51-58) Washington DC.

Hüsemann, B., Lechtenbörger, J., & Vossen, G. (2000). Conceptual data warehouse design. In *Proceedings International Workshop on Design and Management of Data Warehouses*. Stockholm, Sweden.

Kimball, R. (1998). *The data warehouse lifecycle toolkit*. John Wiley & Sons.

Lenz, H. J., & Shoshani, A. (1997). Summarizability in OLAP and statistical databases. In *Proceedings 9th International Conference on Statistical and Scientific Database Management* (pp. 132-143). Washington DC.

Luján-Mora, S., Trujillo, J., & Song, I. Y. (2006). A UML profile for multidimensional modeling in data warehouses. *Data & Knowledge Engineering (DKE)*, 59(3), 725-769.

OMG (2001). *Common Warehouse Metamodel, version 1.0*. February 2001.

Pedersen, T. B., & Jensen, C. (1999). Multidimensional data modeling for complex data. In *Proceedings International Conference on Data Engineering* (pp. 336-345). Sydney, Austrialia.

Rizzi, S. (2006). Conceptual modeling solutions for the data warehouse. In R. Wrembel & C. Koncilia (Eds.), *Data Warehouses and OLAP: Concepts, Architectures and Solutions*, IRM Press, (pp. 1-26).

Sapia, C., Blaschka, M., Hofling, G., & Dinter, B. (1999). Extending the E/R model for the multidimensional paradigm. *Lecture Notes in Computer Science*, 1552, (pp. 105-116).

Simitsis, A., & Vassiliadis P. (2007). *A method for the mapping of conceptual designs to logical blueprints for ETL processes*. Decision Support Systems.

Tryfona, N., Busborg, F., & Borch Christiansen, J. G. (1999). starER: A conceptual model for data warehouse design. In *Proceedings ACM International Workshop on Data Warehousing and OLAP* (pp. 3-8). Kansas City, USA.

Trujillo, J., & Luján-Mora S. (2003). A UML Based Approach for Modeling ETL Processes in Data Warehouses. In *Proceedings ER* (pp. 307-320), Chicago, USA.

Tsois, A., Karayannidis, N., & Sellis, T. (2001). MAC: Conceptual data modeling for OLAP. In *Proceedings International Workshop on Design and Management of Data Warehouses* (pp. 5.1-5.11). Interlaken, Switzerland.

KEY TERMS

Aggregation: The process by which data values are collected with the intent to manage the collection as a single unit.

Conceptual Model: A formalism, with a given expressivity, suited for describing part of the reality, based on some basic constructs and a set of logical and quantitative relationships between them.

DFM: Dimensional Fact Model is a graphical conceptual model specifically devised for describing a multidimensional system.

ETL: The processes of Extracting, Transforming, and Loading data from source data systems into a data warehouse.

Fact: A focus of interest for the decision-making process; typically, it models a set of events occurring in the enterprise world.

Summarizability: The property of correcting summarizing measures along hierarchies by defining a proper operator to compose the measure values characterizing basic events into measure values characterizing aggregated events.

What-If Analysis: What-if analysis is as a data-intensive simulation whose goal is to inspect the behavior of a complex system under some given hypotheses called scenarios.

ENDNOTE

[1] Proposed criteria will be discussed in more details in the main section.

Direction-Aware Proximity on Graphs

Hanghang Tong
Carnegie Mellon University, USA

Yehuda Koren
AT&T Labs - Research, USA

Christos Faloutsos
Carnegie Mellon University, USA

INTRODUCTION

In many graph mining settings, measuring node proximity is a fundamental problem. While most of existing measurements are (implicitly or explicitly) designed for undirected graphs; edge directions in the graph provide a new perspective to proximity measurement: measuring the proximity from A to B; rather than between A and B. (See Figure 1 as an example).

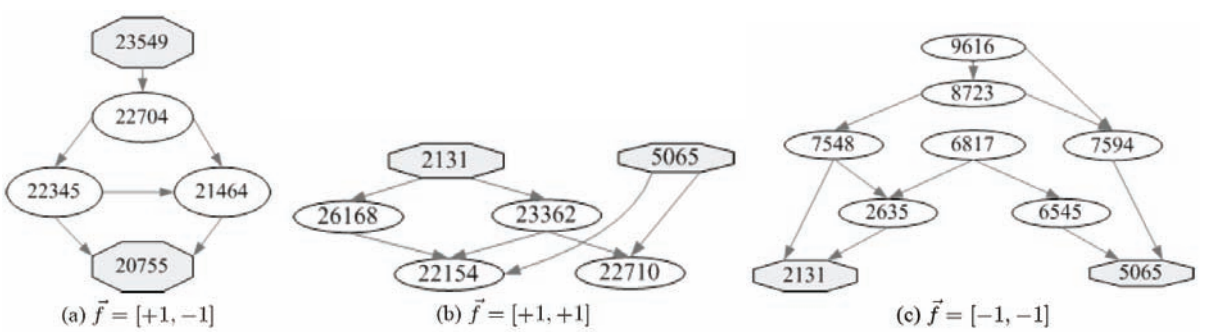
In this chapter, we study the role of edge direction in measuring proximity on graphs. To be specific, we will address the following fundamental research questions in the context of direction-aware proximity:

1. **Problem definitions:** How to define a direction-aware proximity?
2. **Computational issues:** How to compute the proximity score efficiently?
3. **Applications:** How can direction-aware proximity benefit graph mining?

BACKGROUND

In the literature, there are several measures of node proximity. Most standard measures are based on basic graph theoretical concepts -the shortest path length and the maximum flow. However the dependency of these measures on a single element of the graph, the shortest path or the minimum cut, makes them more suitable to managed networks, but inappropriate for measuring the random nature of relationships within social networks or other self organizing networks. Consequently, some works suggested more involved measures such as the sink-augmented delivered current (Faloutsos, Mccurley & Tomkins, 2004), cycle free effective conductance

Figure 1. An example of Dir-CePS. Each node represents a paper and edge denotes 'cited-by' relationship. By employing directional information, Dir-CePS can explore several relationships among the same query nodes (the two octagonal nodes): (a) A query-node to query-node relations; (b) common descendants of the query nodes (paper number 22154 apparently merged the two areas of papers 2131 and 5036); (c) common ancestors of the query nodes: paper 9616 seems to be the paper that initiated the research areas of the query papers/nodes. See 'Main Focus' section for the description of Dir-CePS algorithm.



(Koren, North & Volinsky, 2006), survivable network (Grötschel, Monma & Stoer, 1993), random walks with restart (He, Li, zhang, Tong & Zhang, 2004; Pan, Yang, Faloutsos & Duygulu, 2004). Notice that none of the existing methods meets all the three desirable properties that our approach meets: (a) dealing with directionality, (b) quality of the proximity score and (c) scalability.

Graph proximity is an important building block in many graph mining settings. Representative work includes connection subgraph (Faloutsos, McCurley & Tomkins, 2004; Koren, North & Volinsky, 2006; Tong & Faloutsos 2006), personalized PageRank (Haveliwala, 2002), neighborhood formulation in bipartite graphs (Sun, Qu, Chakrabarti & Faloutsos, 2005), content-based image retrieval (He, Li, zhang, Tong & Zhang, 2004), cross modal correlation discovery (Pan, Yang, Faloutsos & Duygulu, 2004), the BANKS system (Aditya, Bhalotia, Chakrabarti, Hulgeri, Nakhe & Parag 2002), link prediction (Liben-Nowell & Kleinberg, 2003), detecting anomalous nodes and links in the graph (Sun, Qu, Chakrabarti & Faloutsos, 2005), ObjectRank (Balmin, Hristidis & Papakonstantinou, 2004) and RelationalRank (Geerts, Mannila & Terzi, 2004).

MAIN FOCUS

In this Section, we begin by proposing a novel direction-aware proximity definition, based on the notion of escape probability of random walks. It is carefully designed to deal with practical problems such as the inherent noise and uncertainties associated with real life networks. Then, we address computational efficiency, by concentrating on two scenarios: (1) the computation of a single proximity on a large, disk resident graph (with possibly millions of nodes). (2) The computation of multiple pairwise proximities on a medium sized graph (with up to a few tens of thousand nodes). For the former scenario, we develop an iterative solution to avoid matrix inversion, with convergence guarantee. For the latter scenario, we develop an efficient solution, which requires only a single matrix inversion, making careful use of the so-called block-matrix inversion lemma. Finally, we apply our direction-aware proximity to some real life problems. We demonstrate some encouraging results of the proposed direction-aware proximity for predicting the existence of links together with their direction. Another application is so-called "directed center-piece subgraphs."

Direction-Aware Proximity: Definitions

Here we give the main definitions behind our proposed node-to-node proximity measure, namely, the escape probability; then we give the justification for our modifications to it.

Escape Probability

Following some recent works (Faloutsos, McCurley & Tomkins, 2004; Koren, North & Volinsky, 2006; Tong & Faloutsos 2006), our definition is based on properties of random walks associated with the graph. Random walks mesh naturally with the random nature of the self-organizing networks that we deal with here. Importantly, they allow us to characterize relationships based on multiple paths. Random walk notions are known to parallel properties of corresponding electric networks (Doyle & Snell, 1984). For example, this was the basis for the work of Faloutsos et al. (Faloutsos, McCurley & Tomkins, 2004) that measured nodes proximity by employing the notion of effective conductance. Since electric networks are inherently undirected, they cannot be used for our desired directed proximity measure. Nonetheless, the effective conductance can be adequately generalized to handle directional information by using the concept of escape probability (Doyle & Snell, 1984):

DEFINITION 1. The escape probability from node i to node j, $ep_{i,j}$, is the probability that the random particle that starts from node i will visit node j before it returns to node i.

Thus we adopt the escape probability as the starting point for our direction-aware node-to-node proximity score. That is, for the moment, we define the proximity Prox(i, j) from node i to node j as exactly $ep_{i,j}$.

An important quantity for the computation of $ep_{i,j}$ is the *generalized voltage* at each of the nodes, denoted by $v_k(i, j)$: this is defined as the probability that a random particle that starts from node k will visit node j before node i. This way, our proximity measure can be stated as:

$$\text{Prox}(i, j) = ep_{i,j} = \sum_{k=1}^{n} p_{i,k} v_k(i, j) \quad (1)$$

where $p_{i,k}$ is the probability of a direct transition from node i to node k.

For example, in Figure 1(b), we have Prox(A,B)=1 and Prox(B,A)=0.5, which is consistent with our intuition that connections based on longer paths should be weaker. Table 1 lists the main symbols we used in this chapter. Following standard notations, we use capitals for matrices (e.g., W,P), and arrows for column vectors (e.g., $\vec{1}$).

Practical Modifications

Given a weighted directed graph W, there is a natural random walk associated with it whose transition matrix is the normalized version of W, defined as $P = D^{-1}W$. Recall that D is the diagonal matrix of the node out-degrees (specifically, sum of outgoing weights). However, for real problems, this matrix leads to escape probability scores that might not agree with human intuition. In this subsection, we discuss three necessary modifications, to improve the quality of the resulting escape probability scores.

The first modification is to deal with the degree-1 node or dead end. When measuring the escape probability from i to j, we assume that the random particle must eventually reach either i or j. This means that no matter how long it wanders around, it will make unlimited tries, till reaching i or j. This ignores any noise or friction that practically exists in the system and causes the particle to disappear or decay over time. In particular, this problem is manifested in dead-end paths, or degree-1 nodes, which are very common in practical networks whose degree distribution follows a power law. To address this issue, we model the friction in the system by augmenting it with a new node with a zero out degree known as the universal sink (mimicking an absorbing boundary): Now, each node has some small transition probability, 1-c, to reach the sink; and whenever the random particle has reached the sink, it will stay there forever.

The second modification is to deal with the weakly connected pairs. A weakly connected pair is two nodes that are not connected by any directed path, but become connected when considering undirected paths. For such weakly connected pairs, the direction-aware proximity will be zero. However, in some situations, especially when there are missing links in the graph, this might not be desirable. In fact, while we strive to account for

Table 1. Symbols

Symbol	Definition
$W=[w_{i,j}]$	the weighted graph, $1 \le i, j \le n$, $w_{i,j}$ is the weight
A^T	the transpose of matrix A
D	$n \times n$ diagonal matrix of out-degree: $D_{i,i} = \sum_{j=1}^{n} w_{i,j}$ and $D_{i,j} = 0$ for $i \neq j$
$P=[p_{i,j}]$	the transition matrix associated with the graph
$G=[g_{i,j}]$	the G-matrix associated with P: $G = (I - cP)^{-1}$
$\mathbf{V}$	The whole set of the nodes $\mathbf{V} = \{1,2,...,n\}$
$P(i,:)$, $P(:, j)$	i^{th} row of matrix P, and j^{th} column of matrix P, respectively.
Prox(i, j)	node proximity from node i to node j
$\vec{1}$	A column vector whose all elements are 1's
$\vec{e}_i$	A column vector whose i^{th} element is 1 and the rest elements are 0's
c	$(1 - c)$ is the probability to the sink
n	The total number of nodes in the graph

directionality, we also want to consider some portion of the directed link as an undirected one. For example, in phone-call networks, the fact that person A called person B, implies some symmetric relation between the two persons and thereby a greater probability that B will call A (or has already called A, but this link is missing). A random walk modeling of the problem gives us the flexibility to address this issue by introducing lower probability backward edges: whenever we observe an edge $w_{i,j}$ in the original graph, we put another edge in the opposite direction: $w_{j,i} \propto w_{i,j}$. In other words, we replace the original W with $(1-\beta)W + \beta W^{T}$ $(0 < \beta < 1)$.

The third modification is to deal with the size bias. Many of the graphs we deal with are huge, so when computing the proximity, a frequently used technique is to limit its computation to a much smaller candidate graph, which will significantly speed-up the computation. Existing techniques (Faloutsos, McCurley & Tomkins, 2004; Koren, North & Volinsky, 2006), are looking for a moderately sized graph that contains the relevant portions of the network (relatively to a given proximity query), while still enabling quick computation. These techniques can be directly employed when computing our proximity measure. As pointed out in (Koren, North & Volinsky, 2006), for a robust proximity measure, the score given by the candidate graph should not be greater than the score which is based on the full graph. The desired situation is that the proximity between two nodes will monotonically converge to a stable value as the candidate graph becomes larger. However, this is often not the case if we compute the escape probability directly on the candidate graph. To address this issue, we should work with degree preserving candidate graphs. That is, for every node in the candidate graph, its out degree is the same as in the original graph.

Fast Solutions: FastOneDAP and FastAllDAP

Here we deal with the computational aspects of the directed proximity measures defined by equations (2). First, we will show that straight-forward ways to compute these proximities correspond to solving a specific linear system, which involves a matrix inversion. Then, we will propose fast solutions for computing a single proximity on a large graph or all pairwise proximities on a medium sized graph. Notice that these fast solutions will be also beneficial for measuring undirected proximity. For space limitation, we skip the proofs and experimental evaluations of these fast solutions (See (Tong, Koren & Faloutsos, 2007) for details).

Straightforward Solver

Node proximity is based on the linear system (Doyle & Snell, 1984):

$$v_k(i,j) = \sum_{t=1}^{n} c \cdot p_{k,t} \cdot v_t(i,j) \; (k \neq i, j)$$
$$v_i(i,j) = 0; v_j(i,j) = 1 \qquad (2)$$

By solving the above linear system, we have (Doyle & Snell, 1984):

$$\mathrm{Prox}(i,j) = c^2 P(i,\mathbf{I}) G'' P(\mathbf{I},j) + c p_{i,j} \qquad (3)$$

where $\mathbf{I} = \mathbf{V} - \{i, j\}$; $P(i,\mathbf{I})$ is the i^{th} row of matrix P without its i^{th} and j^{th} elements; $P(\mathbf{I}, j)$ is the j^{th} column of matrix P without its i^{th} and j^{th} elements; and $G'' = (I - cP(\mathbf{I},\mathbf{I}))^{-1}$.

In equation (3), the major computational cost is the inversion of an $O(n) \times O(n)$ matrix (G''), which brings up two computational efficiency challenges: (1) For a large graph, say with hundreds of thousands of nodes, matrix inversion would be very slow if not impossible. In this case we would like to completely avoid matrix inversion when computing a single proximity. (2) The above computation requires a different matrix inversion for each proximity value. Thus, it requires performing k separate matrix inversions to compute k proximities. We will show how to eliminate all matrix inversions, but one, regardless of the number of needed proximities.

Computing a Single Proximity

We propose ***FastOneDAP*** (see Box 1), a fast iterative solution for computing one node proximity. ***FastOneDAP*** is $O(mE)$. In other words, ***FastOneDAP*** is linear in the number of edges in the graph. Often, real graphs are very sparse, which means that E is much less than n^2. Thus, ***FastOneDAP*** is significantly more efficient than the straightforward solver, whose running time is $O(n^3)$ due to matrix inversion.

Box 1.

Algorithm 1: FastOneDAP
Input: the transition matrix P, c, the starting node i and the ending node j
Output: the proximity from i to j: Prox(i, j)

1. Initialize:
   a. If $i > j$, $i_0 = i$; else, $i_0 = i - 1$
   b. $\vec{v}^T = \vec{y}^T = \vec{e}_{i_0}^T$
2. Iterate until convergence:
   a. $\vec{y}^T \leftarrow c\vec{y}^T P(\mathbf{V} - \{j\}, \mathbf{V} - \{j\})$
   b. $\vec{v}^T \leftarrow \vec{v}^T + \vec{y}^T$
3. Normalize: $\vec{v}^T = \dfrac{\vec{v}^T}{\vec{v}^T(i_0)}$
4. Return: $\text{Prox}(i, j) = c\vec{v}^T P(\mathbf{V} - \{j\}, j)$

Box 2.

Algorithm 2: FastAllDAP
Input: The transition matrix P, c
Output: All proximities $\text{Pr} = [\text{Prox}(i, j)]_{1 \le i \ne j \le n}$

1. Compute $G = (I - cP)^{-1} = [g_{i,j}]_{1 \le i,j \le n}$
2. For i = 1:n; j = 1:n
   Compute: $\text{Prox}(i, j) = \dfrac{g_{i,j}}{g_{i,i}g_{j,j} - g_{i,j} \cdot g_{j,i}}$

Computing Multiple Proximities

Suppose that we want to compute all $n(n-1)$ pairwise node proximities. There are various situations where one might want to compute all (or many) proximities. First, collecting all proximities and studying their distribution can reveal interesting features of the network and tell us about its global structure. In addition, some algorithms – such as distance based clustering – require the knowledge of all (or at least many) pairwise proximities. Computation of many pairwise proximities in the same network involves solving many linear systems (in fact, one matrix inversion for each proximity). However, there is a lot of redundancy among different linear systems. In fact, we propose a much more efficient method, that need only solve one linear system (or, invert a single matrix) and leverage its result to quickly solve all the others. Consequently, we suggest the ***FastAllDAP*** algorithm (see Box 2). The major benefit of ***FastAllDAP*** is the dramatic reduction of matrix inversion operations from $n(n-1)$ to a single one. In other words, if we treat the matrix inversion as the basic operation, the complexity of ***FastAllDAP*** is O(1).

Applications: Case Studies

We present two applications of our direction-aware proximity as case studies. Here, we only give the basic ideas of these two applications. For the experimental evaluations and some illustrative examples, please refer to (Tong, Koren & Faloutsos, 2007).

Link Prediction

As a proximity measurement, our direction-aware proximity can be directly used for link prediction. More specifically, it can be used for the following two tasks:

T1: (*Existence*) Given two nodes, predict the existence of a link between them

T2: (*Direction*) Given two adjacent (linked) nodes, predict the direction of their link

For T1, we use the simple rule:

A1: Predict a link between i and j iff Prox(i,j) + Prox(j, i) > th (th is a given threshold)

As for directionality prediction, T2, we use the rule:

A2: Predict a link from i to j if Prox(i, j) + Prox(j, i), otherwise predict the opposite direction

Directed Center-Piece Subgraph

The concept connection subgraphs, or center-piece subgraphs, was proposed in (Faloutsos, McCurley & Tomkins, 2004; Tong & Faloutsos, 2006): Given Q query nodes, it creates a subgraph H that shows the relationships between the query nodes. The resulting subgraph should contain the nodes that have strong connection to all or most of the query nodes. Moreover, since this subgraph H is used for visually demonstrating node relations, its visual complexity is capped by setting an upper limit, or a budget on its size. These so-called connection subgraphs (or center-piece subgraphs) were proved useful in various applications, but currently only handle undirected relationships.

With our direction-aware proximity, the algorithm for constructing center-piece subgraphs (CePS) can be naturally generalized to handle directed graphs. A central operation in the original CePS algorithm was to compute an importance score, $r(i, j)$ for a single node j w.r.t. a single query node q_i. Subsequently, these per-query importance scores are combined to importance scores w.r.t. the whole query set, thereby measuring how important each node is relatively to the given group of query nodes. This combination is done through a so-called K_softAND integration that produces $r(\mathbf{Q}, j)$ – the importance score for a single node j w.r.t. the whole query set $\mathbf{Q}$. For more details please refer to (Tong & Faloutsos, 2006).

The main modification that we introduce to the original CePS algorithm is the use of directed proximity for calculating importance scores. The resulting algorithm is named **Dir-CePS** and is given in Algorithm 3 (see Box 3). Directional information must also involve the input to **Dir-CePS** , through the token vector $f = [f_1, ..., f_Q]$ ($f = \pm 1$), which splits the query set into "sources" and "targets", such that each proximity or path are computed from some source to some target. The remaining parts of **Dir-CePS** are exactly the same as in (Tong & Faloutsos, 2006); details are skipped here due to space limitations.

FUTURE TRENDS

We have demonstrated how to encode the edge direction in measuring proximity. Future work in this line includes

Box 3.

Algorithm 3: Dir-CePS
Input: digraph W, query set $\mathbf{Q}$, budget b, token vector $f = [f_1, ..., f_Q]$ ($f = \pm 1$)
Output: the resulting subgraph H

1. For each query node $q_i \in \mathbf{Q}$, each node j in the graph
   If $f_{q_i} = +1$, $r(i, j) = \text{Prox}(q_i, j)$; else $r(i, j) = \text{Prox}(j, q_i)$
2. Combine $r(i, j)$ to get $r(\mathbf{Q}, j)$ by k_SoftAND
3. While H is not big enough
   a. Pick up the node $pd = \arg\max_{j \notin H} r(Q, j)$
   b. For each active source q_i wrt pd
      i. If $f_{q_i} = +1$, find a key path from q_i to pd; else, find a key path from pd to q_i
      ii. Add the key path to H

(1) to generalize the node proximity to measure the relationship between two groups of nodes (see (Tong, Koren & Faloutsos, 2007)); (2) to use parallelism to enable even faster computation of node proximity; and (3) to explore the direction-aware proximity in other data mining tasks.

CONCLUSION

In this work, we study the role of directionality in measuring proximity on graphs. We define a direction-aware proximity measure based on the notion of escape probability. This measure naturally weights and quantifies the multiple relationships which are reflected through the many paths connecting node pairs. Moreover, the proposed proximity measure is carefully designed to deal with practical situations such as accounting for noise and facing partial information.

Given the growing size of networked data, a good proximity measure should be accompanied with a fast algorithm. Consequently we offer fast solutions, addressing two settings. First, we present an iterative algorithm, with convergence guarantee, to compute a single node-to-node proximity value on a large graph. Second, we propose an accurate algorithm that computes all (or many) pairwise proximities on a medium sized graph. These proposed algorithms achieve orders of magnitude speedup compared to straightforward approaches, without quality loss.

Finally, we have studied the applications of the proposed proximity measure to real datasets. Encouraging results demonstrate that the measure is effective for link prediction. Importantly, being direction-aware, it enables predicting not only the existence of the link but also its direction. We also demonstrate how to encode the direction information into the so-called "directed center-piece subgraphs."

REFERENCES

Aditya, B., Bhalotia G., Chakrabarti S., Hulgeri A., Nakhe C., & Parag S. S. (2002). Banks: Browsing and keyword searching in relational databases. In *VLDB*, 1083–1086.

Balmin A., Hristidis V., & Papakonstantinou Y. (2004). Objectrank: Authority-based keyword search in databases. In *VLDB*, 564–575.

Doyle P., & Snell J. (1984). *Random walks and electric networks*. New York: Mathematical Association America.

Faloutsos C., McCurley K. S., & Tomkins A. (2004). Fast discovery of connection subgraphs. In *KDD*, 118–127.

Faloutsos M., Faloutsos P., & Faloutsos C. (1999). On power-law relationships of the internet topology. *SIGCOMM*, 251–262.

Geerts, F., Mannila H., & Terzi E. (2004). Relational link-based ranking. In *VLDB*, 552–563.

Grötschel M., Monma C. L., & Stoer M. (1993). Design of survivable networks. In *Handbooks in operations research and management science 7: Network models*. North Holland.

Haveliwala T. H. (2002). Topic-sensitive pagerank. *WWW*, 517–526.

He J., Li M., Zhang H., Tong H., & Zhang C. (2004). Manifold-ranking based image retrieval. In *ACM Multimedia*, 9–16.

Koren Y., North S. C., & Volinsky C. (2006). Measuring and extracting proximity in networks. In *KDD*, 245–255.

Liben-Nowell D. & J. Kleinberg J. (2003). The link prediction problem for social networks. *Proc. CIKM*, 556–559.

Pan J.-Y., Yang H.-J., Faloutsos C., & Duygulu P. (2004). Automatic multimedia cross-modal correlation discovery. In *KDD*, 653–658.

Sun J., Qu H., Chakrabarti D., & Faloutsos C. (2005). Neighborhood formation and anomaly detection in bipartite graphs. In *ICDM*, 418–425.

Tong H., & Faloutsos C. (2006). Center-piece subgraphs: Problem definition and fast solutions. In *KDD* 404–413.

Tong H., Koren Y., & Faloutsos C. (2007): Fast direction-aware proximity for graph mining. In *KDD*, 747-756

KEY TERMS

Degree Preserving: An operation on the candidate graph, ensuring that the out-degree for each node in the candidate graph is the same as that in the original graph.

Direction Aware Node Proximity: A measure of the closeness from one node to another on the directed graph.

Escape Probability (from node *i* to node *j*): The probability that a random particle that starts from node *i* will visit node *j* before it returns to node *i*.

Generalized Voltage (wrt node *i* to node *j* at node *k*): The probability that a random particle that starts from node *k* will visit node *j* before node *i*

Monotonicity of Proxmity: The property of proximity, measuring that the proximity measured on the smaller candidate graph is always not bigger than that measured on a larger candidate graph.

Transition Matrix: A square matrix, each of whose rows consists of nonnegative real numbers, with each row summing to 1, representing the probability of direct transition between two nodes.

Universal Sink: A node in the graph that has in-link from every other node but no out-link.

Discovering an Effective Measure in Data Mining

Takao Ito
Ube National College of Technology, Japan

INTRODUCTION

One of the most important issues in data mining is to discover an implicit relationship between words in a large corpus and labels in a large database. The relationship between words and labels often is expressed as a function of distance measures. An effective measure would be useful not only for getting the high precision of data mining, but also for time saving of the operation in data mining. In previous research, many measures for calculating the one-to-many relationship have been proposed, such as the complementary similarity measure, the mutual information, and the phi coefficient. Some research showed that the complementary similarity measure is the most effective. The author reviewed previous research related to the measures in one-to-many relationships and proposed a new idea to get an effective one, based on the heuristic approach in this article.

BACKGROUND

Generally, the knowledge discover in databases (KDD) process consists of six stages: data selection, cleaning, enrichment, coding, data mining, and reporting (Adriaans & Zantinge, 1996). Needless to say, data mining is the most important part in the KDD. There are various techniques, such as statistical techniques, association rules, and query tools in a database, for different purposes in data mining. (Agrawal, Mannila, Srikant, Toivonen & Verkamo, 1996; Berland & Charniak, 1999; Caraballo, 1999; Fayyad, Piatetsky-Shapiro, Smyth & Uthurusamy, 1996; Han & Kamber, 2001).

When two words or labels in a large database have some implicit relationship with each other, one of the different purposes is to find out the two relative words or labels effectively. In order to find out relationships between words or labels in a large database, the author found the existence of at least six distance measures after reviewing previously conducted research.

The first one is the mutual information proposed by Church and Hanks (1990). The second one is the confidence proposed by Agrawal and Srikant (1995). The third one is the complementary similarity measure (CSM) presented by Hagita and Sawaki (1995). The fourth one is the dice coefficient. The fifth one is the Phi coefficient. The last two are both mentioned by Manning and Schutze (1999). The sixth one is the proposal measure (PM) suggested by Ishiduka, Yamamoto, and Umemura (2003). It is one of the several new measures developed by them in their paper.

In order to evaluate these distance measures, formulas are required. Yamamoto and Umemura (2002) analyzed these measures and expressed them in four parameters of a, b, c, and d (Table 1).

Suppose that there are two words or labels, x and y, and they are associated together in a large database. The meanings of these parameters in these formulas are as follows:

a. The number of documents/records that have x and y both.
b. The number of documents/records that have x but not y.
c. The number of documents/records that do not have x but do have y.
d. The number of documents/records that do not have either x or y.
n. The total number of parameters a, b, c, and d.

Umemura (2002) pointed out the following in his paper: "Occurrence patterns of words in documents can be expressed as binary. When two vectors are similar, the two words corresponding to the vectors may have some implicit relationship with each other." Yamamoto and Umemura (2002) completed their experiment to test the validity of these indexes under Umemura's con-

Table 1. Kind of distance measures and their formulas

No	Kind of Distance Measures	Formula
1	the mutual information	$I(x_1; y_1) = \log \frac{an}{(a+b)(a+c)}$
2	the confidence	$conf(Y \mid X) = \frac{a}{a+c}$
3	the complementary similarity measure	$S_c(\vec{F}, \vec{T}) = \frac{ad - bc}{\sqrt{(a+c)(b+d)}}$
4	the dice coefficient	$S_d(F, T) = \frac{2a}{(a+b)+(a+c)}$
5	the Phi coefficient	$\varphi_{DE} = \frac{ad - bc}{\sqrt{(a+b)(a+c)(b+d)(c+d)}}$
6	the proposal measure	$S(\vec{F}, \vec{T}) = \frac{a^2 b}{1+c}$

cept. The result of the experiment of distance measures without noisy pattern from their experiment can be seen in Figure 1 (Yamamoto & Umemura, 2002).

The experiment by Yamamoto and Umemura (2002) showed that the most effective measure is the CSM. They indicated in their paper as follows: "All graphs showed that the most effective measure is the complementary similarity measure, and the next is the confidence and the third is asymmetrical average mutual information. And the least is the average mutual information" (Yamamoto and Umemura, 2002). They also completed their experiments with noisy pattern and found the same result (Yamamoto & Umemura, 2002).

MAIN THRUST

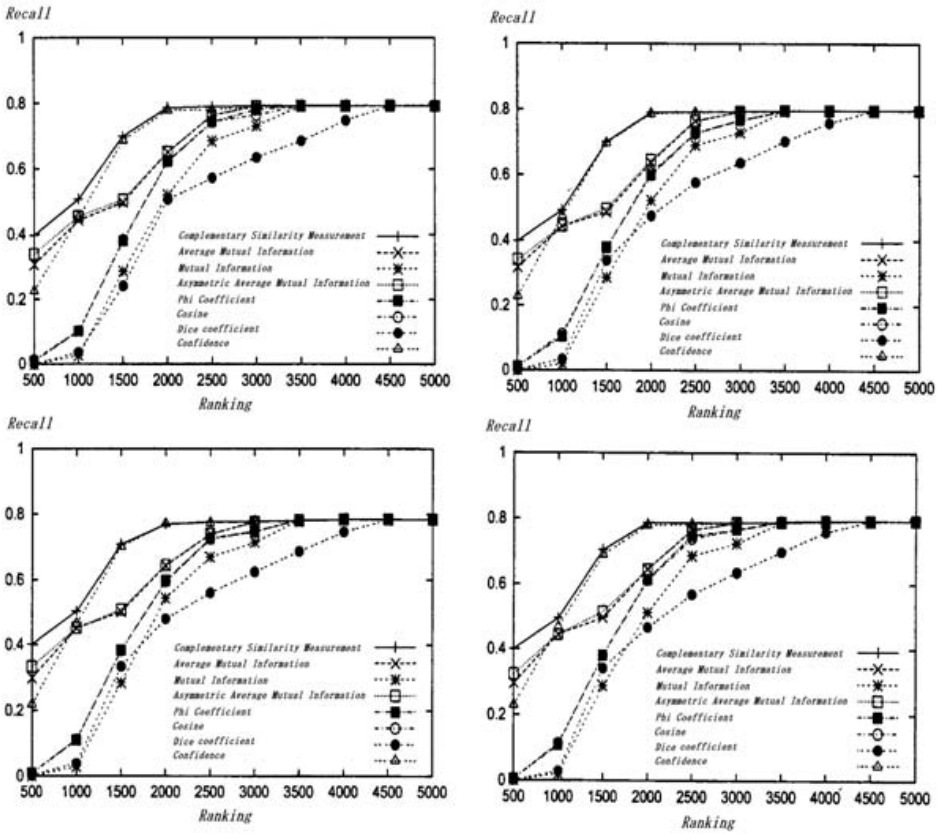
How to select a distance measure is a very important issue, because it has a great influence on the result of data mining (Fayyad, Piatetsky-Shapiro & Smyth, 1996; Glymour, Madigan, Pregibon & Smyth, 1997) The author completed the following three kinds of experiments, based upon the heuristic approach, in order to discover an effective measure in this article (Aho, Kernighan & Weinberger, 1995).

RESULT OF THE THREE KINDS OF EXPERIMENTS

All of these three kinds of experiments are executed under the following conditions. In order to discover an effective measure, the author selected actual data of a place's name, such as the name of prefecture and the name of a city in Japan from the articles of a nationally circulated newspaper, the *Yomiuri*. The reasons for choosing a place's name are as follows: first, there are one-to-many relationships between the name of a prefecture and the name of a city; second, the one-to-many relationship can be checked easily from the maps and telephone directory. Generally speaking, the first name, such as the name of a prefecture, consists of another name, such as the name of a city. For instance, Fukuoka City is geographically located in Fukuoka Prefecture, and Kitakyushu City also is included in Fukuoka Prefecture, so there are one-to-many relationships between the name of the prefecture and the name of the city.

The distance measure would be calculated with a large database in the experiments. The experiments were executed as follows:

Figure 1. Result of the experiment of distance measures without noisy pattern



Step 1. Establish the database of the newspaper.
Step 2. Choose the prefecture name and city name from the database mentioned in step 1.
Step 3. Count the number of parameters a, b, c, and d from the newspaper articles.
Step 4. Then, calculate the distance measure adopted.
Step 5. Sort the result calculated in step 4 in descent order upon the distance measure.
Step 6. List the top 2,000 from the result of step 5.
Step 7. Judge the one-to-many relationship whether it is correct or not.
Step 8. List the top 1,000 as output data and count its number of correct relationships.
Step 9. Finally, an effective measure will be found from the result of the correct number.

To uncover an effective one, two methods should be considered. The first one is to test various combinations of each variable in distance measure and to find the best combination, depending upon the result. The second one is to assume that the function is a stable one and that only part of the function can be varied.

The first one is the experiment with the PM.

The total number of combinations of five variables is 3,125. The author calculated all combinations, except for the case when the denominator of the PM becomes zero. The result of the top 20 in the PM experiment, using a year's amount of articles from the *Yomiuri* in 1991, is calculated as follows.

In Table 2, the No. 1 function of a11c1 has a highest correct number, which means

$$S(\vec{F},\vec{T}) = \frac{a \times 1 \times 1}{c+1} = \frac{a}{1+c}.$$

The No. 11 function of 1a1c0 means

$$S(\vec{F},\vec{T}) = \frac{1 \times a \times 1}{c+0} = \frac{a}{c}.$$

The rest function in Table 2 can be read as mentioned previously.

This result appears to be satisfactory. But to prove whether it is satisfactory or not, another experiment should be done. The author adopted the Phi coefficient and calculated its correct number. To compare with the result of the PM experiment, the author completed the experiment of iterating the exponential index of denominator in the Phi coefficient from 0 to 2, with 0.01 step based upon the idea of fractal dimension in the complex theory instead of fixed exponential index at 0.5. The result of the top 20 in this experiment, using a year's amount of articles from the *Yomiuri* in 1991, just like the first experiment, can be seen in Table 3.

Compared with the result of the PM experiment mentioned previously, it is obvious that the number of correct relationships is less than that of the PM experiment; therefore, it is necessary to uncover a new, more effective measure. From the previous research done by Yamamoto and Umemura (2002), the author found that an effective measure is the CSM.

The author completed the third experiment using the CMS. The experiment began by iterating the exponential index of the denominator in the CSM from 0 to 1, with 0.01 steps just like the second experiment. Table 4 shows the result of the top 20 in this experiment, using a year's amount of articles from the *Yomiuri* from 1991-1997.

Table 2. Result of the top 20 in the PM experiment

No	Function	Correct Number	No	Function	Correct Number
1	a11c1	789	11	1a1c0	778
2	a111c	789	12	1a10c	778
3	1a1c1	789	13	11acc	778
4	1a11c	789	14	11ac0	778
5	11ac1	789	15	11a0c	778
6	11a1c	789	16	aa1cc	715
7	a11cc	778	17	aa1c0	715
8	a11c0	778	18	aa10c	715
9	a110c	778	19	a1acc	715
10	1a1cc	778	20	a1ac0	715

Table 3. Result of the top 20 in the Phi coefficient experiment

No	Exponential Index	Correct Number	No	Exponential index	Correct Number
1	0.26	499	11	0.20	465
2	0.25	497	12	0.31	464
3	0.27	495	13	0.19	457
4	0.24	493	14	0.32	445
5	0.28	491	15	0.18	442
6	0.23	488	16	0.17	431
7	0.29	480	17	0.16	422
8	0.22	478	18	0.15	414
9	0.21	472	19	0.14	410
10	0.30	470	20	0.13	403

It is obvious by these results that the CSM is more effective than the PM and the Phi coefficient. The relationship of the complete result of the exponential index of the denominator in the CSM and the correct number can be seen as in Figure 2.

The most effective exponential index of the denominator in the CSM is from 0.73 to 0.85, but not as 0.5, as many researchers believe. It would be hard to get the best result with the usual method using the CSM.

Determine the Most Effective Exponential Index of the Denominator in the CSM

To discover the most effective exponential index of the denominator in the CSM, a calculation about the relationship between the exponential index and the total number of documents of n was carried out, but none was found. In fact, it is hard to consider that the exponential index would vary with the difference of the number of documents.

The details of the four parameters of a, b, c, and d in Table 4 are listed in Table 5.

The author adopted the average of the cumulative exponential index to find the most effective exponential index of the denominator in the CSM. Based upon the results in Table 4, calculations for the average of the cumulative exponential index of the top 20 and those results are presented in Figure 3.

From the results in Figure 3, it is easy to understand that the exponential index converges at a constant value of about 0.78. Therefore, the exponential index could be fixed at a certain value, and it will not vary with the size of documents.

In this article, the author puts the exponential index at 0.78 to forecast the correct number. The gap between the calculation result forecast by the revised method and the maximum result of the third experiment is illustrated in Figure 4.

FUTURE TRENDS

Many indexes have been developed for the discovery of an effective measure to determine an implicit relationship between words in a large corpus or labels in a large database. Ishiduka, Yamamoto, and Umemura (2003) referred to part of them in their research paper. Almost all of these approaches were developed by a variety of mathematical and statistical techniques, a conception of neural network, and the association rules.

For many things in the world, part of the economic phenomena obtain a normal distribution referred to in statistics, but others do not, so the author developed the CSM measure from the idea of fractal dimension in this article. Conventional approaches may be inherently inaccurate, because usually they are based upon

Figure 2. Relationships between the exponential indexes of denominator in the CSM and the correct number

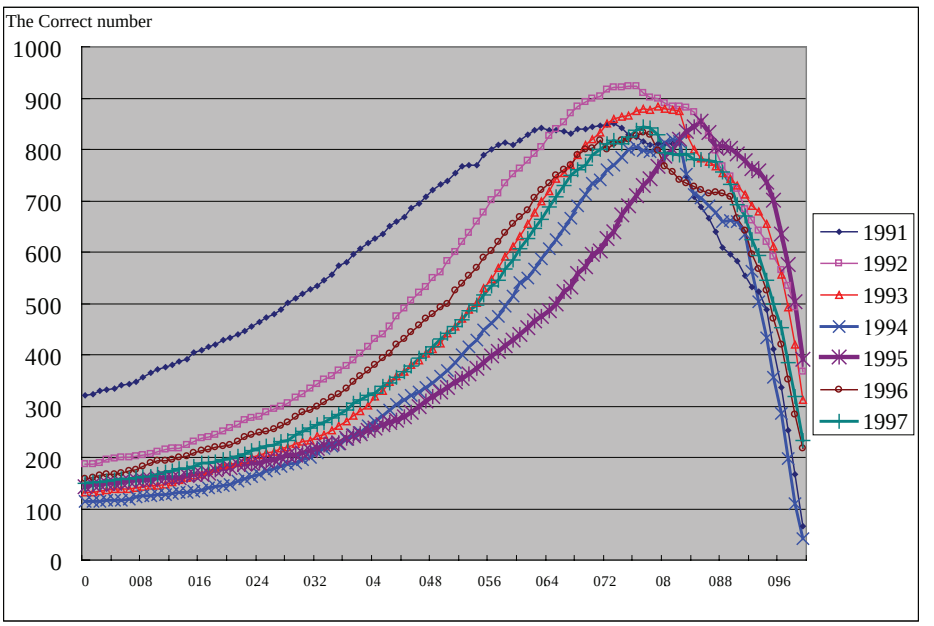


Table 4. Result of the top 20 in the CSM experiment

1991		1992		1993		1994		1995		1996		1997	
E.I.	C.N.	E.I.	C.N.	E.I.	C.N.	E.I.	C.N.	E.I.	C.N.	E.I.	C.N.	E.I.	C.N.
0.73	850	0.75	923	0.79	883	0.81	820	0.85	854	0.77	832	0.77	843
0.72	848	0.76	922	0.80	879	0.82	816	0.84	843	0.78	828	0.78	842
0.71	846	0.74	920	0.77	879	0.8	814	0.83	836	0.76	826	0.76	837
0.70	845	0.73	920	0.81	878	0.76	804	0.86	834	0.75	819	0.79	829
0.74	842	0.72	917	0.78	878	0.75	800	0.82	820	0.74	818	0.75	824
0.63	841	0.77	909	0.82	875	0.79	798	0.88	807	0.71	818	0.73	818
0.69	840	0.71	904	0.76	874	0.77	798	0.87	805	0.73	812	0.74	811
0.68	839	0.78	902	0.75	867	0.78	795	0.89	803	0.70	803	0.72	811
0.65	837	0.79	899	0.74	864	0.74	785	0.81	799	0.72	800	0.71	801
0.64	837	0.70	898	0.73	859	0.73	769	0.90	792	0.69	800	0.80	794
0.62	837	0.69	893	0.72	850	0.72	761	0.80	787	0.79	798	0.81	792
0.66	835	0.80	891	0.71	833	0.83	751	0.91	777	0.68	788	0.83	791
0.67	831	0.81	884	0.83	830	0.71	741	0.79	766	0.67	770	0.82	790
0.75	828	0.68	884	0.70	819	0.70	734	0.92	762	0.80	767	0.70	789
0.61	828	0.82	883	0.69	808	0.84	712	0.93	758	0.66	761	0.84	781
0.76	824	0.83	882	0.83	801	0.69	711	0.78	742	0.81	755	0.85	780
0.60	818	0.84	872	0.68	790	0.85	706	0.94	737	0.65	749	0.86	778
0.77	815	0.67	870	0.85	783	0.86	691	0.77	730	0.82	741	0.87	776
0.58	814	0.66	853	0.86	775	0.68	690	0.76	709	0.83	734	0.69	769
0.82	813	0.85	852	0.67	769	0.87	676	0.95	701	0.64	734	0.68	761

Table 5. Relationship between the maximum correct number and their parameters

Year	Exponential Index	Correct Number	a	b	c	d	n
1991	0.73	850	2,284	46,242	3,930	1,329	53,785
1992	0.75	923	453	57,636	1,556	27	59,672
1993	0.79	883	332	51,649	1,321	27	53,329
1994	0.81	820	365	65,290	1,435	36	67,126
1995	0.85	854	1,500	67,914	8,042	190	77,646
1996	0.77	832	636	56,529	2,237	2,873	62,275
1997	0.77	843	520	69,145	3,033	56	72,754

Figure 3. Relationships between the exponential index and the average of the number of the cumulative exponential index

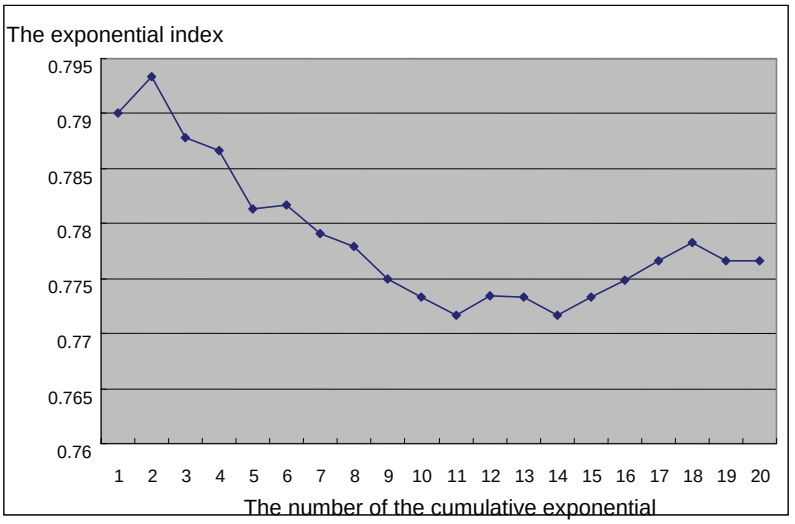
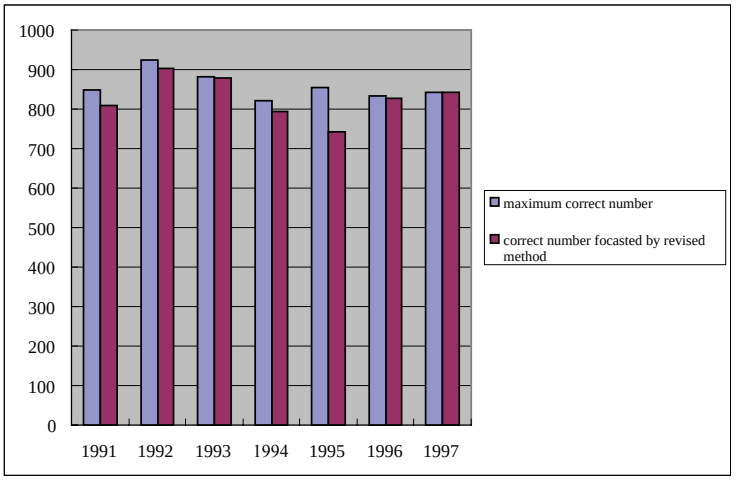


Figure 4. Gaps between the maximum correct number and the correct number forecasted by the revised meth-



linear mathematics. The events of a concurrence pattern of correlated pair words may be explained better by nonlinear mathematics. Typical tools of nonlinear mathematics are complex theories, such as chaos theory, cellular automaton, percolation model, and fractal theory. It is not hard to predict that many more new measures will be developed in the near future, based upon the complex theory.

CONCLUSION

Based upon previous research of the distance measures, the author discovered an effective measure of distance measure in one-to-many relationships with the heuristic approach. Three kinds of experiments were conducted, and it was confirmed that an effective measure is the CSM. In addition, it was discovered that the most effective exponential of the denominator in the CSM is 0.78, not 0.50, as many researchers believe.

A great deal of work still needs to be carried out, and one of them is the meaning of the 0.78. The meaning of the most effective exponential of denominator 0.78 in the CSM should be explained and proved mathematically, although its validity has been evaluated in this article.

ACKNOWLEDGMENTS

The author would like to express his gratitude to the following: Kyoji Umemura, Professor, Toyohashi University of Technology; Taisuke Horiuchi, Associate Professor, Nagano National College of Technology; Eiko Yamamoto, Researcher, Communication Research Laboratories; Yuji Minami, Associate Professor, Ube National College of Technology; Michael Hall, Lecturer, Nakamura-Gakuen University; who suggested a large number of improvements, both in content and computer program.

REFERENCES

Adriaans, P., & Zantinge, D. (1996). *Data mining*. Addison Wesley Longman Limited.

Agrawal, R., & Srikant, R. (1995). Mining of association rules between sets of items in large databases. *Proceedings of the ACM SIGMOD Conference on Management of Data*.

Agrawal, R. et al. (1996). *Fast discovery of association rules, advances in knowledge discovery and data mining*. Cambridge, MA: MIT Press.

Aho, A.V., Kernighan, B.W., & Weinberger, P.J. (1989). *The AWK programming language*. Addison-Wesley Publishing Company, Inc.

Berland, M., & Charniak, E. (1999). Finding parts in very large corpora. *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*.

Caraballo, S.A. (1999). Automatic construction of a hypernym-labeled noun hierarchy from text. *Proceedings of the Association for Computational Linguistics*.

Church, K.W., & Hanks, P. (1990). Word association norms, mutual information, and lexicography. *Computational Linguistics, 16*(1), 22-29.

Fayyad, U.M. et al. (1996). *Advances in knowledge discovery and data mining*.AAAI Press/MIT Press.

Fayyad, U.M., Piatetsky-Shapiro, G., & Smyth, P. (1996). The KDD process for extracting useful knowledge from volumes of data. *Communications of the ACM*, *39*(11), 27-34.

Glymour, C. et al. (1997). Statistical themes and lessons for data mining. *Data Mining and Knowledge Discover*, *1*(1), 11-28.

Hagita, N., & Sawaki, M. (1995). Robust recognition of degraded machine-printed characters using complimentary similarity measure and error-correction learning. *Proceedings of the SPIE—The International Society for Optical Engineering*.

Han, J., & Kamber, M. (2001). *Data mining*. Morgan Kaufmann Publishers.

Ishiduka, T., Yamamoto, E., & Umemura, K. (2003). *Evaluation* of a function presumes the one-to-many relationship [unpublished research paper] [Japanese edition]).

Manning, C.D., & Schutze, H. (1999). *Foundations of statistical natural language processing*. MIT Press.

Umemura, K. (2002). Selecting the most highly correlated pairs within a large vocabulary. *Proceedings of the COLING Workshop SemaNet'02, Building and Using Semantic Networks*.

Yamamoto, E., & Umemura, K. (2002). A similarity measure for estimation of one-to-many relationship in corpus [Japanese edition]. *Journal of Natural Language Processing*, *9*, 45-75.

KEY TERMS

Complementary Similarity Measurement: An index developed experientially to recognize a poorly printed character by measuring the resemblance of the correct pattern of the character expressed in a vector. The author calls this a diversion index to identify the one-to-many relationship in the concurrence patterns of words in a large corpus or labels in a large database in this article.

Confidence: An asymmetric index that shows the percentage of records for which A occurred within the group of records and for which the other two, X and

Y, actually occurred under the association rule of X, Y => A.

Correct Number: For example, if the city name is included in the prefecture name geographically, the author calls it correct. So, in this index, the correct number indicates the total number of correct one-to-many relationship calculated on the basis of the distance measures.

Distance Measure: One of the calculation techniques to discover the relationship between two implicit words in a large corpus or labels in a large database from the viewpoint of similarity.

Mutual Information: Shows the amount of information that one random variable x contains about another y. In other words, it compares the probability of observing x and y, together with the probabilities of observing x and y independently.

Phi Coefficient: One metric for corpus similarity measured upon the chi square test. It is an index to calculate the frequencies of the four parameters of a, b, c, and d in documents based upon the chi square test with the frequencies expected for independence.

Proposal Measure: An index to measure the concurrence frequency of the correlated pair words in documents, based upon the frequency of two implicit words that appear. It will have high value, when the two implicit words in a large corpus or labels in a large database occur with each other frequently.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 364-371, copyright 2005 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

Discovering Knowledge from XML Documents

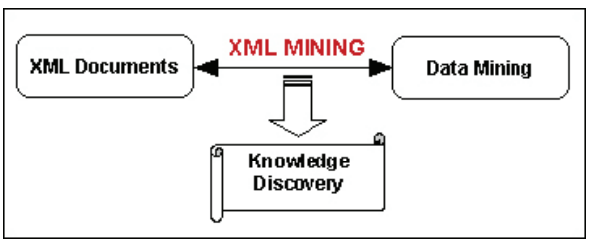
Richi Nayak
Queensland University of Technology, Australia

INTRODUCTION

XML is the new standard for information exchange and retrieval. An XML document has a schema that defines the data definition and structure of the XML document (Abiteboul et al., 2000). Due to the wide acceptance of XML, a number of techniques are required to retrieve and analyze the vast number of XML documents. Automatic deduction of the structure of XML documents for storing semi-structured data has been an active subject among researchers (Abiteboul et al., 2000; Green et al., 2002). A number of query languages for retrieving data from various XML data sources also has been developed (Abiteboul et al., 2000; W3c, 2004). The use of these query languages is limited (e.g., limited types of inputs and outputs, and users of these languages should know exactly what kinds of information are to be accessed). Data mining, on the other hand, allows the user to search out unknown facts, the information hidden behind the data. It also enables users to pose more complex queries (Dunham, 2003).

Figure 1 illustrates the idea of integrating data mining algorithms with XML documents to achieve knowledge discovery. For example, after identifying similarities among various XML documents, a mining technique can analyze links between tags occurring together within the documents. This may prove useful in the analysis of e-commerce Web documents recommending personalization of Web pages.

Figure 1. XML mining scheme



BACKGROUND: WHAT IS XML MINING?

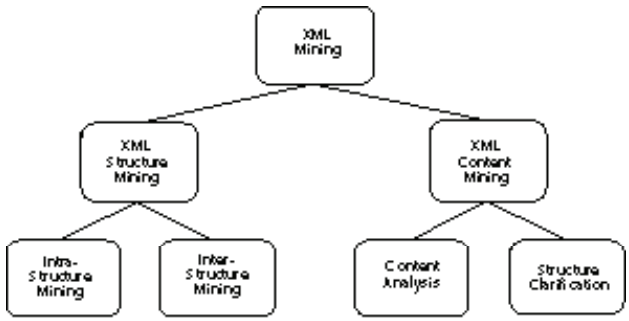
XML mining includes mining of structures as well as contents from XML documents, depicted in Figure 2 (Nayak et al., 2002). Element tags and their nesting therein dictate the structure of an XML document (Abiteboul et al., 2000). For example, the textual structure enclosed by <author>… </author> is used to describe the author tuple and its corresponding text in the document. Since XML provides a mechanism for tagging names with data, knowledge discovery on the semantics of the documents becomes easier for improving document retrieval on the Web. Mining of XML structure is essentially mining of schema including intrastructure mining, and interstructure mining.

Intrastructure Mining

Concerned with the structure within an XML document. Knowledge is discovered about the internal structure of XML documents in this type of mining. The following mining tasks can be applied.

The classification task of data mining maps a new XML document to a predefined class of documents. A schema is interpreted as a description of a class of XML documents. The classification procedure takes a collection of schemas as a training set and classifies new XML documents according to this training set.

Figure 2. A taxonomy of XML mining



The clustering task of data mining identifies similarities among various XML documents. A clustering algorithm takes a collection of schemas to group them together on the basis of self-similarity. These similarities are then used to generate new schema. As a generalization, the new schema is a superclass to the training set of schemas. This generated set of clustered schemas can now be used in classifying new schemas. The superclass schema also can be used in integration of heterogeneous XML documents for each application domain. This allows users to find, collect, filter, and manage information sources more effectively on the Internet.

The association data mining describes relationships between tags that tend to occur together in XML documents that can be useful in the future. By transforming the tree structure of XML into a pseudo-transaction, it becomes possible to generate rules of the form "if an XML document contains a <craft> tag, then 80% of the time it also will contain a <licence> tag." Such a rule then may be applied in determining the appropriate interpretation for homographic tags.

Interstructure Mining

Concerned with the structure between XML documents. Knowledge is discovered about the relationship between subjects, organizations, and nodes on the Web in this type of mining. The following mining tasks can be applied.

Clustering schemas involves identifying similar schemas. The clusters are used in defining hierarchies of schemas. The schema hierarchy overlaps instances on the Web, thus discovering authorities and hubs (Garofalakis et al. 1999). Creators of schema are identified as authorities, and creators of instances are hubs. Additional mining techniques are required to identify all instances of schema present on the Web. The following application of classification can identify the most likely places to mine for instances. Classification is applied with namespaces and URIs (Uniform Resource Identifiers). Having previously associated a set of schemas with a particular namespace or URI, this information is used to classify new XML documents originating from these places.

Content is the text between each start and end tag in XML documents. Mining for XML content is essentially mining for values (an instance of a relation), including content analysis and structural clarification.

Content Analysis

Concerned with analysing texts within XML documents. The following mining tasks can be applied to contents.

Classification is performed on XML content, labeling new XML content as belonging to a predefined class. To reduce the number of comparisons, pre-existing schemas classify the new document's schema. Then, only the instance classifications of the matching schemas need to be considered in classifying a new document.

Clustering on XML content identifies the potential for new classifications. Again, consideration of schemas leads to quicker clustering; similar schemas are likely to have a number of value sets. For example, all schemas concerning vehicles have a set of values representing cars, another set representing boats, and so forth. However, schemas that appear dissimilar may have similar content. Mining XML content inherits some problems faced in text mining and analysis. Synonymy and polysemy can cause difficulties, but the tags surrounding the content usually can help resolve ambiguities.

Structural Clarification

Concerned with distinguishing the similar structured documents based on contents. The following mining tasks can be performed.

Content provides support for alternate clustering of similar schemas. Two distinctly structured schemas may have document instances with identical content. Mining these avails new knowledge. Vice versa, schemas provide support for alternate clustering of content. Two XML documents with distinct content may be clustered together, given that their schemas are similar.

Content also may prove important in clustering schemas that appear different but have instances with similar content. Due to heterogeneity, the incidence of synonyms is increased. Are separate schemas actually describing the same thing, only with different terms? While thesauruses are vital, it is impossible for them to be exhaustive for the English language, let alone handle all languages. Conversely, schemas appearing similar actually are completely different, given homographs. The similarity of the content does not distinguish the semantic intention of the tags. Mining, in this case, pro-

vides probabilities of a tag having a particular meaning or a relationship between meaning and a URI.

METHODS OF XML STRUCTURE MINING

Mining of structures from a well-formed or valid document is straightforward, since a valid document has a schema mechanism that defines the syntax and structure of the document. However, since the presence of schema is not mandatory for a well-formed XML document, the document may not always have an accompanying schema. To describe the semantic structure of the documents, schema extraction tools are needed to generate schema for the given well-formed XML documents.

DTD Generator (Kay, 2000) generates the DTD for a given XML document. However, the DTD generator yields a distinct DTD for every XML document; hence, a set of DTDs is defined for a collection of XML documents rather than an overall DTD. Thus, the application of data mining operations will be difficult in this matter. Tools such as XTRACT (Garofalakis, 2000) and DTD-Miner (Moh et al., 2000) infer an accurate and semantically meaningful DTD schema for a given collection of XML documents. However, these tools depend critically on being given a relatively homogeneous collection of XML documents. In such heterogeneous and flexible environment as the Web, it is not reasonable to assume that XML documents related to the same topic have the same document structure.

Due to a number of limitations using DTDs as an internal structure, such as limited set of data types, loose structure constraints, and limitation of content to textual, many researchers propose the extraction of XML schema as an extension to XML DTDs (Feng et al., 2002; Vianu, 2001). In Chidlovskii (2002), a novel XML schema extraction algorithm is proposed, based on the Extended Context-Free Grammars (ECFG) with a range of regular expressions. Feng et al. (2002) also presented a semantic network-based design to convey the semantics carried by the XML hierarchical data structures of XML documents and to transform the model into an XML schema. However, both of these proposed algorithms are very complex.

Mining of structures from ill-formed XML documents (that lack any fixed and rigid structure) are performed by applying the structure extraction approaches developed for semi-structured documents. But not all of these techniques can effectively support the structure extraction from XML documents that is required for further application of data mining algorithms. For instance, the NoDoSe tool (Adelberg & Denny, 1999) determines the structure of a semi-structured document and then extracts the data. This system is based primarily on plain text and HTML files, and it does not support XML. Moreover, in Green, et al. (2002), the proposed extraction algorithm considers both structure and contents in semi-structured documents, but the purpose is to query and build an index. They are difficult to use without some alteration and adaptation for the application of data mining algorithms.

An alternative method is to approach the document as the Object Exchange Model (OEM) (Nestorov et al. 1999; Wang et al. 2000) data by using the corresponding data graph to produce the most specific data guide (Nayak et al., 2002). The data graph represents the interactions between the objects in a given data domain. When extracting a schema from a data graph, the goal is to produce the most specific schema graph from the original graph. This way of extracting schema is more general than using the schema for a guide, because most of the XML documents do not have a schema, and sometimes, if they have a schema, they do not conform to it.

METHODS OF XML CONTENT MINING

Before knowledge discovery in XML documents occurs, it is necessary to query XML tags and content to prepare the XML material for mining. An SQL-based query can extract data from XML documents. There are a number of query languages, some specifically designed for XML and some for semi-structured data, in general. Semi-structured data can be described by the grammar of SSD (semi-structured data) expressions. The translation of XML to SSD expression is easily automated (Abiteboul et al., 2000). Query languages for semi-structured data exploit path expressions. In this way, data can be queried to an arbitrary depth. Path expressions are elementary queries with the results returned as a set of nodes. However, the ability to return results as semi-structured data is required, which path expressions alone cannot do. Combining path expressions with SQL-style syntax provides greater flexibility in testing for equality, performing

joins, and specifying the form of query results. Two such languages are Lorel (Abiteboul et al., 2000) and Unstructured Query Language (UnQL) (Farnandez et al., 2000). UnQL requires more precision and is more reliant on path expressions.

XML-QL, XML-GL, XSL, and Xquery are designed specifically for querying XML (W3c, 2004). XML-QL (Garofalsaki et al., 1999) and Xquery bring together regular path expressions, SQL-style query techniques, and XML syntax. The great benefit is the construction of the result in XML and, thus, transforming XML data from one schema to another. Extensible Stylesheet Language (XSL) is not implemented as a query language but is intended as a tool for transforming XML to HTML. However, XSL'_S select pattern is a mechanism for information retrieval and, as such, is akin to a query (W3c, 2004). XML-GL (Ceri et al., 1999) is a graphical language for querying and restructuring XML documents.

FUTURE TRENDS

There has been extensive effort to devise new technologies to process and integrate XML documents, but a lot of open possibilities still exist. For example, integration of data mining, XML data models and database languages will increase the functionality of relational database products, data warehouses, and XML products. Also, to satisfy the range of data mining users (from naive to expert users), future work should include mining user graphs that are structural information of Web usages, as well as visualization of mined data. As data mining is applied to large semantic documents or XML documents, extraction of information should consider rights management of shared data. XML mining should have the authorization level to empower security to restrict only appropriate users to discover classified information.

CONCLUSION

XML has proved effective in the process of transmitting and sharing data over the Internet. Companies want to bring this advantage into analytical data, as well. As XML material becomes more abundant, the ability to gain knowledge from XML sources decreases due to their heterogeneity and structural irregularity; the idea behind the XML data mining looks like a solution to put to work. Using XML data in the mining process has become possible through new Web-based technologies that have been developed. Simple Object Access Protocol (SOAP) is a new technology that has enabled XML to be used in data mining. For example, vTag Web Mining Server aims at monitoring and mining of the Web with the use of information agents accessed by SOAP (vtag, 2003). Similarly, XML for Analysis defines a communication structure for an application programming interface, which aims at keeping client programming independent from the mechanics of data transport but, at the same time, providing adequate information concerning the data and ensuring that it is properly handled (XMLanalysis, 2003). Another development, YALE, is an environment for machine learning experiments that uses XML files to describe data mining experiments setup (Yale, 2004). The Data Miner's ARCADE also uses XML as the target language' for all data mining tools within its environment (Arcade, 2004).

REFERENCES

Abiteboul, S., Buneman, P., & Suciu, D. (2000). *Data on the Web: From relations to semistructured data and XML*. San Francisco, CA: Morgan Kaumann.

Adelberg, B., & Denny, M. (1999). Nodose version 2.0. *Proceedings of the ACM SIGMOD Conference on Management of Data*, Seattle, Washington.

Arcade. (2004). *http://datamining.csiro.au/arcade.html*

Ceri, S. et al. (1999). XML—Gl: A graphical language for querying and restructuring XML documents. *Proceedings of the 8th International WWW Conference*, Toronto, Canada.

Chidlovskii, B. (2002). Schema extraction from XML collections. *Proceedings of the 2nd ACM/IEEE-CS Joint Conference on Digital Libraries*, Portland, Oregon.

Dunham, M.H. (2003). *Data mining: Introductory and advanced topics.* Upper Saddle River, NJ: Prentice Hall

Farnandez, M., Buneman, P., & Suciu, D. (2000). UNQL: A query language and algebra for semistructured

data based on structural recursion. *VLDB JOURNAL: Very Large Data Bases, 9*(1), 76-110.

Feng, L., Chang, E., & Dillon, T. (2002). A semantic network-based design methodology for XML documents. *ACM Transactions of Information Systems (TOIS), 20*(4), 390-421.

Garofalakis, M. et al. (1999). Data mining and the Web: Past, present and future. *Proceedings of the Second International Workshop on Web Information and Data Management*, Kansas City, Missouri.

Garofalakis, M.N. et al. (2000). XTRACT: A system for extracting document type descriptors from XML documents. *Proceedings of the 2000 ACM SIGMOD International Conference on Management* of Data, Dallas, Texas.

Green, R., Bean, C.A., & Myaeng, S.H. (2002). *The semantics of relationships: An interdisciplinary perspective*. Boston: Kluwer Academic Publishers.

Kay, M. (2000). SAXON DTD generator—A tool to generate XML DTDs. Retrieved January 2, 2003, from *http://home.iclweb.com/ic2/mhkay/dtdgen.html*

Moh, C.-H., & Lim, E.-P. (2000). DTD-miner: A tool for mining DTD from XML documents. *Proceedings of the Second International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems*, California.

Nayak, R., Witt, R., & Tonev, A. (2002, June). Data mining and XML documents. *Proceedings of the 2002 International Conference on Internet Computing*, Nevada.

Nestorov, S. et al. (1999). Representative objects: Concise representation of semi-structured, hierarchical data. *Proceedings of the IEEE Proc on Management of Data*, Seattle, Washington.

Vianu, V. (2001). A Web odyssey: from Codd to XML. *Proceedings of the 20th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, California.

Vtag. (2003). *http://www.connotate.com/csp.as*p

W3c. (2004). XML query (Xquery). Retrieved March 18, 2004, from *http://www.w3c.org/XML/Query*

Wang, Q., Yu, X.J., & Wong, K. (2000). Approximate graph scheme extraction for semi-structured data. *Proceedings of the 7th International Conference on Extending Database Technology*, Konstanz.

XMLanalysis. (2003). *http://www.intelligenteai.com/feature/011004/editpage.shtml*

Yale. (2004). *http://yale.cs.uni-dortmund.de/*

KEY TERMS

Ill-Formed XML Documents: Lack any fixed and rigid structure.

Valid XML Document: To be valid, an XML document additionally must conform (at least) to an explicitly associated document schema definition.

Well-Formed XML Documents: To be well-formed, a page's XML must have properly nested tags, unique attributes (per element), one or more elements, exactly one root element, and a number of schema-related constraints. Well-formed documents may have a schema, but they do not conform to it.

XML Content Analysis Mining: Concerned with analysing texts within XML documents.

XML Interstructure Mining: Concerned with the structure between XML documents. Knowledge is discovered about the relationship among subjects, organizations, and nodes on the Web.

XML Intrastructure Mining: Concerned with the structure within an XML document(s). Knowledge is discovered about the internal structure of XML documents.

XML Mining: Knowledge discovery from XML documents (heterogeneous and structural irregular). For example, clustering data mining techniques can group a collection of XML documents together according to the similarity of their structures. Classification data mining techniques can classify a number of heterogeneous XML documents into a set of predefined classification of schemas to improve XML document handling and achieve efficient searches.

XML Structural Clarification Mining: Concerned with distinguishing the similar structured documents based on contents.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 372-376, copyright 2005 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

Discovering Unknown Patterns in Free Text

Jan H Kroeze
University of Pretoria, South Africa

Machdel C Matthee
University of Pretoria, South Africa

INTRODUCTION

A very large percentage of business and academic data is stored in textual format. With the exception of metadata, such as author, date, title and publisher, this data is not overtly structured like the standard, mainly numerical, data in relational databases. Parallel to data mining, which finds new patterns and trends in numerical data, text mining is the process aimed at discovering unknown patterns in free text. Owing to the importance of competitive and scientific knowledge that can be exploited from these texts, "text mining has become an increasingly popular and essential theme in data mining" (Han & Kamber, 2001, p. 428).

Text mining is an evolving field and its relatively short history goes hand in hand with the recent explosion in availability of electronic textual information. Chen (2001, p. vi) remarks that "text mining is an emerging technical area that is relatively unknown to IT professions". This explains the fact that despite the value of text mining, most research and development efforts still focus on data mining using structured data (Fan et al., 2006).

In the next section, the background and need for text mining will be discussed after which the various uses and techniques of text mining are described. The importance of visualisation and some critical issues will then be discussed followed by some suggestions for future research topics.

BACKGROUND

Definitions of text mining vary a great deal, from views that it is an advanced form of information retrieval (IR) to those that regard it as a sibling of data mining:

- Text mining is the discovery of texts.
- Text mining is the exploration of available texts.
- Text mining is the extraction of information from text.
- Text mining is the discovery of new knowledge in text.
- Text mining is the discovery of new patterns, trends and relations in and among texts.

Han & Kamber (2001, pp. 428-435), for example, devote much of their rather short discussion of text mining to information retrieval. However, one should differentiate between text mining and information retrieval. Text mining does not consist of searching through metadata and full-text databases to find existing information. The point of view expressed by Nasukawa & Nagano (2001, p. 969), to wit that text mining "is a text version of generalized data mining", is correct. Text mining should "focus on finding valuable patterns and rules in text that indicate trends and significant features about specific topics" (ibid., p. 967).

Like data mining, text mining is a proactive process that automatically searches data for new relationships and anomalies to serve as a basis for making business decisions aimed at gaining competitive advantage (cf. Rob & Coronel, 2004, p. 597). Although data mining can require some interaction between the investigator and the data-mining tool, it can be considered as an automatic process because "*data-mining tools automatically search the data for anomalies and possible relationships, thereby identifying problems that have not yet been identified by the end user*", while mere data analysis "*relies on the end users to define the problem, select the data, and initiate the appropriate data analyses to generate the information that helps model and solve problems those end-users uncover*" (ibid.). The same distinction is valid for text mining. Therefore, text-mining tools should also "*initiate* analyses to create knowledge" (ibid., p. 598).

In practice, however, the borders between data analysis, information retrieval and text mining are not always quite so clear. Montes-y-Gómez et al. (2004)

proposed an integrated approach, called *contextual exploration*, which combines robust access (IR), non-sequential navigation (hypertext) and content analysis (text mining).

THE NEED FOR TEXT MINING

Text mining can be used as an effective business intelligence tool for gaining competitive advantage through the discovery of critical, yet hidden, business information. As a matter of fact, all industries traditionally rich in documents and contracts can benefit from text mining (McKnight, 2005). For example, in medical science, text mining is used to build and structure medical knowledge bases, to find undiscovered relations between diseases and medications or to discover gene interactions, functions and relations (De Bruijn & Martin, 2002, p. 8). A recent application of this is where Gajendran, Lin and Fyhrie (2007) use text mining to predict potentially novel target genes for osteoporosis research that has not been reported on in previous research. Also, government intelligence and security agencies find text mining useful in predicting and preventing terrorist attacks and other security threats (Fan et al., 2006).

USES OF TEXT MINING

The different types of text mining have the following in common: it differs from data mining in that it extracts patterns from free (natural language) text rather than from structured databases. However, it does this by using data mining techniques: "it numericizes the unstructured text document and then, using data mining tools and techniques, extracts patterns from them" (Delen and Crossland, 2007:4). In this section various uses of text mining will be discussed, as well as the techniques employed to facilitate these goals. Some examples of the implementation of these text mining approaches will be referred to. The approaches that will be discussed include categorisation, clustering, concept-linking, topic tracking, anomaly detection and web mining.

Categorisation

Categorisation focuses on identifying the main themes of a document after which the document is grouped according to these. Two techniques of categorisation are discussed below:

Keyword-Based Association Analysis

Association analysis looks for correlations between texts based on the occurrence of related keywords or phrases. Texts with similar terms are grouped together. The pre-processing of the texts is very important and includes parsing and stemming, and the removal of words with minimal semantic content. Another issue is the problem of compounds and non-compounds - should the analysis be based on singular words or should word groups be accounted for? (cf. Han & Kamber, 2001, p. 433). Kostoff et al. (2002), for example, have measured the frequencies and proximities of phrases regarding electrochemical power to discover central themes and relationships among them. This knowledge discovery, combined with the interpretation of human experts, can be regarded as an example of knowledge creation through intelligent text mining.

Automatic Document Classification

Electronic documents are classified according to a predefined scheme or training set. The user compiles and refines the classification parameters, which are then used by a computer program to categorise the texts in the given collection automatically (cf. Sullivan, 2001, p. 198). Classification can also be based on the analysis of collocation ("the juxtaposition or association of a particular word with another particular word or words" (The Oxford Dictionary, 9th Edition, 1995)). Words that often appear together probably belong to the same class (Lopes et al., 2004). According to Perrin & Petry (2003) "useful text structure and content can be systematically extracted by collocational lexical analysis" with statistical methods. Text classification can be applied by businesses, for example, to personalise B2C e-commerce applications. Zhang and Jiao (2007) did this by using a model that anticipates customers' heterogeneous requirements as a pre-defined scheme for the classification of e-commerce sites for this purpose (Zhang & Jiao, 2007).

Clustering

Texts are grouped according to their own content into categories that were not previously known. The documents are analysed by a clustering computer program, often a neural network, but the clusters still have to be interpreted by a human expert (Hearst, 1999). Document pre-processing (tagging of parts of speech, lemmatisation, filtering and structuring) precedes the actual clustering phase (Iiritano et al., 2004). The clustering program finds similarities between documents, e.g. common author, same themes, or information from common sources. The program does not need a training set or taxonomy, but generates it dynamically (cf. Sullivan, 2001, p. 201). One example of the use of text clustering in the academic field is found in the work of Delen and Crossland (2007) whose text-mining tool processes articles from three major journals in the management information systems field to identify major themes and trends of research in this area.

Concept Linking

In text databases, concept linking is the finding of meaningful, high levels of correlations between text entities. Concept linking is implemented by the technique called link analysis, "the process of building up networks of interconnected objects through relationships in order to expose patterns and trends" (Westphal, 1998, p. 202). The user can, for example, suggest a broad hypothesis and then analyse the data in order to prove or disprove this hunch. It can also be an automatic or semi-automatic process, in which a surprisingly high number of links between two or more nodes may indicate relations that have hitherto been unknown. Link analysis can also refer to the use of algorithms to build and exploit networks of hyperlinks in order to find relevant and related documents on the Web (Davison, 2003). Concept linking is used, for example, to identify experts by finding and evaluating links between persons and areas of expertise (Ibrahim, 2004). Yoon & Park (2004) use concept linking and information visualisation to construct a visual network of patents, which facilitates the identification of a patent's relative importance: "The coverage of the application is wide, ranging from new idea generation to ex post facto auditing" (ibid, p. 49).

Topic Tracking

Topic tracking is the discovery of a developing trend in politics or business, which may be used to predict recurring events. It thus involves the discovery of patterns that are related to time frames, for example, the origin and development of a news thread (cf. Montes-y-Gómez et al., 2001). The technique that is used to do this is called sequence analysis. A sequential pattern is the arrangement of a number of elements, in which the one leads to the other over time (Wong et al., 2000). An example of topic tracking is a system that remembers user profiles and, based on that, predict other documents of interest to the user (Delen & Crossland, 2007).

Anomaly Detection

Anomaly detection is the finding of information that violates the usual patterns, e.g. a book that refers to a unique source, or a document lacking typical information. Link analysis and keyword-based analysis, referred to above, are techniques that may also be used for this purpose. An example of anomaly detection is the detection of irregularities in news reports or different topic profiles in newspapers (Montes-y-Gómez et al., 2001).

Web Mining

"Text mining is about looking for patterns in natural language text…. Web mining is the slightly more general case of looking for patterns in hypertext and often applies graph theoretical approaches to detect and utilise the structure of web sites." (New Zealand Digital Library, 2002)

In addition to the obvious hypertext analysis, various other techniques are used for web mining. Marked-up language, especially XML tags, facilitates text mining because the tags can often be used to simulate database attributes and to convert data-centric documents into databases, which can then be exploited (Tseng & Hwung, 2002). Mark-up tags also make it possible to create "artificial structures [that] help us understand the relationship between documents and document components" (Sullivan, 2001, p. 51). Such tags could, for example, be used to store linguistic analyses regarding the various language modules of a text, enabling

the application of data warehousing and data mining concepts in the humanities (cf. Kroeze, 2007).

Web-applications nowadays integrate a variety of types of data, and web mining will focus increasingly on the effective exploitation of such multi-faceted data. Web mining will thus often include an integration of various text mining techniques. One such an application of web mining where multiple text mining techniques are used is natural language queries or "question answering" (Q&A). Q&A deals with finding the best answer to a given question on the web (Fan et al., 2006). Another prominent application area of web mining is recommendation systems (e.g. personalisation), the design of which should be robust since security is becoming an increasing concern (Nasraoui et al., 2005).

INFORMATION VISUALISATION

Information visualisation "puts large textual sources in a visual hierarchy or map and provides browsing capabilities, in addition to simple searching" (Fan et al., 2006:80). Although this definition categorises information visualisation as an information retrieval technique and aid, it is often referred to as visual text mining (Fan et al., 2006; Lopes et al., 2007). A broader understanding of information visualisation therefore includes the use of visual techniques for not only information retrieval but also to interpret the findings of text mining efforts. According to Lopes et al. (2007), information visualisation reduces the complexity of text mining by helping the user to build more adequate cognitive models of trends and the general structure of a complex set of documents.

CRITICAL ISSUES

Many sources on text mining refer to text as "unstructured data". However, it is a fallacy that text data are unstructured. Text is actually highly structured in terms of morphology, syntax, semantics and pragmatics. On the other hand, it must be admitted that these structures are not directly visible: "… text represents factual information … in a complex, rich, and opaque manner" (Nasukawa & Nagano, 2001, p. 967).

Authors also differ on the issue of natural language processing within text mining. Some prefer a more statistical approach (cf. Hearst, 1999), while others feel that linguistic parsing is an essential part of text mining. Sullivan (2001, p. 37) regards the representation of meaning by means of syntactic-semantic representations as essential for text mining: "Text processing techniques, based on morphology, syntax, and semantics, are powerful mechanisms for extracting business intelligence information from documents…. We can scan text for meaningful phrase patterns and extract key features and relationships". According to De Bruijn & Martin (2002, p. 16), "[l]arge-scale statistical methods will continue to challenge the position of the more syntax-semantics oriented approaches, although both will hold their own place."

In the light of the various definitions of text mining, it should come as no surprise that authors also differ on what qualifies as text mining and what does not. Building on Hearst (1999), Kroeze, Matthee & Bothma (2003) use the parameters of novelty and data type to distinguish between information retrieval, standard text mining and intelligent text mining (see Figure 1).

Halliman (2001, p. 7) also hints at a scale of newness of information: "Some text mining discussions stress the importance of 'discovering new knowledge.' And the new knowledge is expected to be new to everybody. From a practical point of view, we believe that business text should be 'mined' for information that is 'new enough' to give a company a competitive edge once the information is analyzed."

Another issue is the question of when text mining can be regarded as "intelligent". Intelligent behavior is "the ability to learn from experience and apply knowledge acquired from experience, handle complex situations, solve problems when important information is missing, determine what is important, react quickly and correctly to a new situation, understand visual images, process and manipulate symbols, be creative and imaginative, and use heuristics" (Stair & Reynolds, 2001, p. 421). Intelligent text mining should therefore refer to the interpretation and evaluation of discovered patterns.

FUTURE RESEARCH

Mack and Hehenberger (2002, p. S97) regards the automation of "human-like capabilities for comprehending complicated knowledge structures" as one of the frontiers of "text-based knowledge discovery". Incorporating more artificial intelligence abilities into text-mining tools will facilitate the transition from

Figure 1. A differentiation between information retrieval, standard and intelligent metadata mining, and standard and intelligent text mining (abbreviated from Kroeze, Matthee & Bothma, 2003)

Novelty level: / Data type:	Non-novel investigation: Information retrieval	Semi-novel investigation: Knowledge discovery	Novel investigation: Knowledge creation
Metadata (overtly structured)	Information retrieval of metadata	Standard metadata mining	Intelligent metadata mining
Free text (covertly structured)	Information retrieval of full texts	Standard text mining	Intelligent text mining

mainly statistical procedures to more intelligent forms of text mining. Fan et al. (2006) consider duo-mining as an important future consideration. This involves the integration of data mining and text mining into a single system and will enable users to consolidate information by analyzing both structured data from databases as well as free text from electronic documents and other sources.

CONCLUSION

Text mining can be regarded as the next frontier in the science of knowledge discovery and creation, enabling businesses to acquire sought-after competitive intelligence, and helping scientists of all academic disciplines to formulate and test new hypotheses. The greatest challenges will be to select and integrate the most appropriate technology for specific problems and to popularise these new technologies so that they become instruments that are generally known, accepted and widely used.

REFERENCES

Chen, H. (2001). *Knowledge management systems: A text mining perspective*. Tucson, AZ: University of Arizona.

Davison, B.D. (2003). *Unifying text and link analysis*. Paper read at the Text-Mining & Link-Analysis Workshop of the 18th International Joint Conference on Artificial Intelligence. Retrieved November 8, 2007, from http://www-2.cs.cmu.edu/~dunja/TextLink2003/

De Bruijn, B. & Martin, J. (2002). Getting to the (c)ore of knowledge: Mining biomedical literature. *International Journal of Medical Informatics*, *67*(1-3), 7-18.

Delen, D., & Crossland, M.D. (2007). Seeding the survey and analysis of research literature with text mining. *Expert Systems with Applications*, *34*(3), 1707-1720.

Fan, W., Wallace, L., Rich, S., & Zhang, Z. (2006). Tapping the power of text mining. *Communications of the ACM*, *49*(9), 77-82.

Gajendran, V.K., Lin, J., & Fyhrie, D.P. (2007). An application of bioinformatics and text mining to the discovery of novel genes related to bone biology. *Bone*, *40*, 1378-1388.

Halliman, C. (2001). *Business intelligence using smart techniques: Environmental scanning using text mining and competitor analysis using scenarios and manual simulation*. Houston, TX: Information Uncover.

Han, J. & Kamber, M. (2001). *Data mining: Concepts and techniques*. San Francisco: Morgan Kaufmann.

Hearst, M.A. (1999, June 20-26). Untangling text data mining. In *Proceedings of ACL'99: the 37th Annual Meeting of the Association for Computational Linguistics*, University of Maryland. Retrieved August 2, 2002, from http://www.ai.mit.edu/people/jimmylin/papers/Hearst99a.pdf.

Ibrahim, A. (2004). Expertise location: Can text mining help? In N.F.F. Ebecken, C.A. Brebbia & A. Zanasi (Eds.) *Data Mining IV* (pp. 109-118). Southampton UK: WIT Press.

Iiritano, S., Ruffolo, M. & Rullo, P. (2004). Preprocessing method and similarity measures in clustering-based

text mining: A preliminary study. In N.F.F. Ebecken, C.A. Brebbia & A. Zanasi (Eds.) *Data Mining IV* (pp. 73-79). Southampton UK: WIT Press.

Kostoff, R.N., Tshiteya, R., Pfeil, K.M. & and Humenik, J.A. (2002). Electrochemical power text mining using bibliometrics and database tomography. *Journal of Power Sources, 110*(1), 163-176.

Kroeze, J.H., Matthee, M.C. & Bothma, T.J.D. (2003, 17-19 September). Differentiating data- and text-mining terminology. In J. Eloff, P. Kotzé, A. Engelbrecht & M. Eloff (eds.) *IT Research in Developing Countries: Proceedings of the Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists (SAICSIT 2003)*(pp. 93-101). Fourways, Pretoria: SAICSIT.

Kroeze, J.H. (2007, May 19-23). Round-tripping Biblical Hebrew linguistic data. In M. Khosrow-Pour (Ed.) *Managing Worldwide Operations and Communications with Information Technology: Proceedings of 2007 Information Resources Management Association, International Conference, Vancouver, British Columbia, Canada,* (IRMA 2007) (pp. 1010-1012). Hershey, PA: IGI Publishing.

Lopes, A.A., Pinho, R., Paulovich, F.V. & Minghim, R. (2007). Visual text mining using association rules. *Computers & Graphics, 31*, 316-326.

Lopes, M.C.S., Terra, G.S., Ebecken, N.F.F. & Cunha, G.G. (2004). Mining text databases on clients opinion for oil industry. In N.F.F. Ebecken, C.A. Brebbia & A. Zanasi (Eds.) *Data Mining IV* (pp. 139-147). Southampton, UK: WIT Press.

Mack, R. & Hehenberger, M. (2002). Text-based knowledge discovery: Search and mining of life-science documents. *Drug Discovery Today, 7*(11), S89-S98.

McKnight, W. (2005, January). Building business intelligence: Text data mining in business intelligence. *DM Review*. Retrieved November 8, 2007, from http://www.dmreview.com/article_sub.cfm?articleId=1016487

Montes-y-Gómez, M., Gelbukh, A. & López-López, A. (2001 July-September). Mining the news: Trends, associations, and deviations. *Computación y Sistemas, 5*(1). Retrieved November 8, 2007, from http://ccc.inaoep.mx/~mmontesg/publicaciones/2001/NewsMining-CyS01.pdf

Montes-y-Gómez, M., Pérez-Coutiño, M., Villaseñor-Pineda, L. & López-López, A. (2004). *Contextual exploration of text collections* (LNCS, 2945). Retrieved November 8, 2007, from http://ccc.inaoep.mx/~mmontesg/publicaciones/2004/ContextualExploration-CICLing04.pdf

Nasraoui, O., Zaiane, O.R., Spiliopoulou, M., Mobasher, B., Masand, B., Yu, P.S. 2005. WebKDD 2005: Web mining and web usage analysis post-workshop report. *SIGKDD Explorations, 7*(2), 139-142.

Nasukawa, T. & Nagano, T. (2001). Text analysis and knowledge mining system. *IBM Systems Journal, 40*(4), 967-984.

New Zealand Digital Library, University of Waikato. (2002). *Text mining*. Retrieved November 8, 2007, from http://www.cs.waikato.ac.nz/~nzdl/textmining/

Perrin, P. & Petry, F.E. (2003). Extraction and representation of contextual information for knowledge discovery in texts. *Information Sciences, 151,* 125-152.

Rob, P. & Coronel, C. (2004). *Database systems: design, implementation, and management, 6th ed.* Boston: Course Technology.

Stair, R.M. & Reynolds, G.W. (2001). *Principles of information systems: a managerial approach, 5th ed.* Boston: Course Technology.

Sullivan, D. (2001). *Document warehousing and text mining: Techniques for improving business operations, marketing, and sales*. New York: John Wiley

Tseng, F.S.C. & Hwung, W.J. (2002). An automatic load/extract scheme for XML documents through object-relational repositories. *Journal of Systems and Software, 64*(3), 207-218.

Westphal, C. & Blaxton, T. (1998). *Data mining solutions: Methods and tools for solving real-world problems*. New York: John Wiley.

Wong, P.K., Cowley, W., Foote, H., Jurrus, E. & Thomas, J. (2000). Visualizing sequential patterns for text mining. In *Proceedings of the IEEE Symposium on Information Visualization 2000,* 105. Retrieved November 8, 2007, from http://portal.acm.org/citation.cfm

Yoon, B. & Park, Y. (2004). A text-mining-based patent network: Analytical tool for high-technology trend. *The Journal of High Technology Management Research, 15*(1), 37-50.

Zhang, Y., & Jiao, J. (2007). An associative classification-based recommendation system for personalization in B2C e-commerce applications. *Expert Systems with Applications*, 33, 357-367.

KEY TERMS

Business Intelligence: "Any information that reveals threats and opportunities that can motivate a company to take some action" (Halliman, 2001, p. 3).

Competitive Advantage: The head start a business has owing to its access to new or unique information and knowledge about the market in which it is operating.

Hypertext: A collection of texts containing links to each other to form an interconnected network (Sullivan, 2001, p. 46).

Information Retrieval: The searching of a text collection based on a user's request to find a list of documents organised according to its relevance, as judged by the retrieval engine (Montes-y-Gómez et al., 2004). Information retrieval should be distinguished from text mining.

Knowledge Creation: The evaluation and interpretation of patterns, trends or anomalies that have been discovered in a collection of texts (or data in general), as well as the formulation of its implications and consequences, including suggestions concerning reactive business decisions.

Knowledge Discovery: The discovery of patterns, trends or anomalies that already exist in a collection of texts (or data in general), but have not yet been identified or described.

Mark-Up Language: Tags that are inserted in free text to mark structure, formatting and content. XML tags can be used to mark attributes in free text and to transform free text into an exploitable database (cf. Tseng & Hwung, 2002).

Metadata: Information regarding texts, e.g. author, title, publisher, date and place of publication, journal or series, volume, page numbers, key words, etc.

Natural Language Processing (NLP): The automatic analysis and/or processing of human language by computer software, "focussed on understanding the contents of human communications". It can be used to identify relevant data in large collections of free text for a data mining process (Westphal & Blaxton, 1998, p. 116).

Parsing: A (NLP) process that analyses linguistic structures and breaks them down into parts, on the morphological, syntactic or semantic level.

Stemming: Finding the root form of related words, for example singular and plural nouns, or present and past tense verbs, to be used as key terms for calculating occurrences in texts.

Text Mining: The automatic analysis of a large text collection in order to identify previously unknown patterns, trends or anomalies, which can be used to derive business intelligence for competitive advantage or to formulate and test scientific hypotheses.

Discovery Informatics from Data to Knowledge

William W. Agresti
Johns Hopkins University, USA

INTRODUCTION

It is routine to hear and read about the information explosion, how we are all overwhelmed with data and information. Is it progress when our search tools report that our query resulted in 300,000 hits? Or, are we still left to wonder where is the information that we really wanted? How far down the list must we go to find it?

Discovery informatics is a distinctly 21st century emerging methodology that brings together several threads of research and practice aimed at making sense out of massive data sources. It is defined as "the study and practice of employing the full spectrum of computing and analytical science and technology to the singular pursuit of discovering new information by identifying and validating patterns in data" (Agresti, 2003).

BACKGROUND

The rapid rise in the amount of information generated each year may be quite understandable. After all, the world's population is growing, and countries like China and India, with very large populations, are becoming increasingly influential worldwide. However, the real reason why people are confronted with so much more information in their lives and work is that the information has real benefits for them. However, these benefits are not always or often realized, and therein lay the motivation for discovery informatics.

Companies today are data mining with more highly granular data to better understand their customers' buying habits. As a result, there is pressure on all businesses to attain the same level of understanding or be left behind – and being left behind in the 21st century can mean going out of business. Not-for-profits are becoming equally adept at mining data to discover which likely donors are most cost-effective to cultivate. Increasing granularity enables more targeted marketing, but with more data requiring more analysis. A co-conspirator in this infoglut is the declining cost to store the data. Organizations don't need to make choices on what data to keep. They can keep it all.

The task of making sense out of this burgeoning mass of data is growing more difficult every day. Effectively transforming this data into usable knowledge is the challenge of discovery informatics. In this broad-based conceptualization, discovery informatics may be seen as taking shape by drawing on more established disciplines:

- **Data analysis and visualization:** analytic frameworks, interactive data manipulation tools, visualization environments
- **Database management:** data models, data analysis, data structures, data management, federation of databases, data warehouses, database management systems
- **Pattern recognition:** statistical processes, classifier design, image data analysis, similarity measures, feature extraction, fuzzy sets, clustering algorithms
- **Information storage and retrieval:** indexing, content analysis, abstracting, summarization, electronic content management, search algorithms, query formulation, information filtering, relevance and recall, storage networks, storage technology
- **Knowledge management:** knowledge sharing, knowledge bases, tacit and explicit knowledge, relationship management, content structuring, knowledge portals, collaboration support systems
- **Artificial intelligence:** learning, concept formation, neural nets, knowledge acquisition, intelligent systems, inference systems, Bayesian methods, decision support systems, problem solving, intelligent agents, text analysis, natural language processing

What distinguishes discovery informatics is that it brings coherence across dimensions of technologies and domains to focus on discovery. It recognizes and builds upon excellent programs of research and practice in individual disciplines and application areas. It looks selectively across these boundaries to find anything (e.g., ideas, tools, strategies, and heuristics) that will help with the critical task of discovering new information.

To help characterize discovery informatics, it may be useful to see if there are any roughly analogous developments elsewhere. Two examples, knowledge management and core competence, may be instructive as reference points.

Knowledge management, which began its evolution in the early 1990s, is "the practice of transforming the intellectual assets of an organization into business value" (Agresti, 2000). Of course, before 1990 organizations, to varying degrees, knew that the successful delivery of products and services depended on the collective knowledge of employees. However, KM challenged organizations to focus on knowledge and recognize its key role in their success. They found value in addressing questions such as:

- What is the critical knowledge that should be managed?
- Where is the critical knowledge?
- How does knowledge get into products and services?

When C. K. Prahalad and Gary Hamel published their highly influential paper, "The Core Competence of the Corporation," (Prahalad and Hamel, 1990) companies had some capacity to identify what they were good at. However, as with KM, most organizations did not appreciate how identifying and cultivating core competencies (CC) may make the difference between competitive or not. A core competence is not the same as "what you are good at" or "being more vertically integrated." It takes dedication, skill, and leadership to effectively identify, cultivate, and deploy core competences for organizational success.

Both KM and CC illustrate the potential value of taking on a specific perspective. By doing so, an organization will embark on a worthwhile re-examination of familiar topics: its customers, markets, knowledge sources, competitive environment, operations, and success criteria. The claim of this chapter is that discovery informatics represents a distinct perspective; one that is potentially highly beneficial because, like KM and CC, it strikes at what is often an essential element for success and progress, discovery.

Embracing the concept of strategic intelligence for an organization, Liebowitz (2006) has explored the relationships and synergies among knowledge management, business intelligence, and competitive intelligence.

MAIN THRUST OF THE CHAPTER

This section will discuss the common elements of discovery informatics and how it encompasses both the technology and application dimensions.

Common Elements of Discovery Informatics

The only constant in discovery informatics is data and an interacting entity with an interest in discovering new information from it. What varies, and has an enormous effect on the ease of discovering new information, is everything else, notably:

- Data:
  - Volume: How much?
  - Accessibility: Ease of access and analysis?
  - Quality: How clean and complete is it? Can it be trusted as accurate?
  - Uniformity: How homogeneous is the data? Is it in multiple forms, structures, formats, and locations?
  - Medium: Text, numbers, audio, video, image, electronic or magnetic signals or emanations?
  - Structure of the data: Formatted rigidly, partially, or not at all? If text, does it adhere to known language?
- Interacting Entity:
  - Nature: Is it a person or intelligent agent?
  - Need, question, or motivation of the user: What is prompting a person to examine this data? How sharply defined is the question or need? What expectations exist about what might be found? If the motivation is to find

something interesting, what does that mean in context?

A wide variety of activities may be seen as variations on the scheme above. An individual using a search engine to search the Internet for a home mortgage serves as an instance of query-driven discovery. A retail company looking for interesting patterns in its transaction data is engaged in data-driven discovery. An intelligent agent examining newly posted content to the Internet based on a person's profile is also engaged in a process of discovery, only the interacting entity is not a person.

Discovery across Technologies

The technology dimension is considered broadly to include automated hardware and software systems, theories, algorithms, architectures, techniques, methods, and practices. Included here are familiar elements associated with data mining and knowledge discovery, such as clustering, link analysis, rule induction, machine learning, neural networks, evolutionary computation, genetic algorithms, and instance based learning (Wang, 2003).

However, the discovery informatics viewpoint goes further, to activities and advances that are associated with other areas but should be seen as having a role in discovery. Some of these activities, like searching or knowledge sharing, are well known from everyday experience.

Conducting searches on the Internet is a common practice that needs to be recognized as part of a thread of information retrieval. Because it is practiced essentially by all Internet users and it involves keyword search, there is a tendency to minimize its importance. Search technology is extremely sophisticated (Henzinger, 2007).

Searching the Internet has become a routine exercise in information acquisition. The popularity of searching is a testament to the reality of the myriad of complex relationships among concepts, putting an end to any hope that some simple (e.g., tree-like) structure of the Internet will ever be adequate (see, e.g., Weinberger, 2007).

While finding the "right" information by searching is a challenging task, the view from the content-producing side of the Internet reveals its own vexing questions. Businesses want desperately to know how to get their sites to show up in the results of Internet searches. The importance of search results to business is evidenced by the frequent search engine marketing conferences conducted regularly around the world. People use the web to find products and services; "customers vote with their mice" (Moran, 2008). If your company's offerings can emerge from the search process on page one, without having to pay for the privilege, the competitive advantage is yours.

People always have some starting point for their searches. Often it is not a keyword, but instead is a concept. So people are forced to perform the transformation from a notional concept of what is desired to a list of one or more keywords. The net effect can be the familiar many-thousand "hits" from the search engine. Even though the responses are ranked for relevance (itself a rich and research-worthy subject), people may still find that the returned items do not match their intended concepts.

Offering hope for improved search are advances in concept-based search (Houston and Chen, 2004), more intuitive with a person's sense of "Find me content like this," where this can be a concept embodied in an entire document or series of documents. For example, a person may be interested in learning which parts of a new process guideline are being used in practice in the pharmaceutical industry. Trying to obtain that information through keyword searches would typically involve trial and error on various combinations of keywords. What the person would like to do is to point a search tool to an entire folder of multimedia electronic content and ask the tool to effectively integrate over the folder contents and then discover new items that are similar. Current technology can support this ability to associate a "fingerprint" with a document (Heintze, 2004), to characterize its meaning, thereby enabling concept-based searching. Discovery informatics recognizes that advances in search and retrieval enhance the discovery process.

With more multimedia available online, the discovery process extends to finding content from videos, sound, music, and images to support learning. There is an active research community exploring issues such as the best search parameters, performance improvements, and retrieval based on knowing fragments of both text and other media (see, e.g., Rautiainen, Ojala & Seppänen, 2004).

This same semantic analysis can be exploited in other settings, such as within organizations. It is possible

now to have your email system prompt you based on the content of messages you compose. When you click "Send," the email system may open a dialogue box, "Do you also want to send that to Mary?" The system has analyzed the content of your message, determining that, for messages in the past having similar content, you have also sent them to Mary. So the system is now asking you if you have perhaps forgotten to include her. While this feature can certainly be intrusive and bothersome unless it is wanted, the point is that the same semantic analysis advances are at work here as with the Internet search example.

The "informatics" part of discovery informatics also conveys the breadth of science and technology needed to support discovery. There are commercially available computer systems and special-purpose software dedicated to knowledge discovery (e.g., see listings at http://www.kdnuggets.com/). The informatics support includes comprehensive hardware-software discovery platforms as well as advances in algorithms and data structures, which are core subjects of computer science. The latest developments in data-sharing, application integration, and human-computer interfaces are used extensively in the automated support of discovery. Particularly valuable, because of the voluminous data and complex relationships, are advances in visualization. Commercial visualization packages are widely used to display patterns and enable expert interaction and manipulation of the visualized relationships. There are many dazzling displays on the Internet that demonstrate the benefits of more effective representations of data; see, for example, the visualizations at (Smashing Magazine, 2007).

Discovery across Domains

Discovery informatics encourages a view that spans application domains. Over the past decade, the term has been most often associated with drug discovery in the pharmaceutical industry, mining biological data. The financial industry also was known for employing talented programmers to write highly sophisticated mathematical algorithms for analyzing stock trading data, seeking to discover patterns that could be exploited for financial gain. Retailers were prominent in developing large data warehouses that enabled mining across inventory, transaction, supplier, marketing, and demographic databases. The situation -- marked by drug discovery informatics, financial discovery informatics, et al. -- was evolving into one in which discovery informatics was preceded by more and more words as it was being used in an increasing number of domain areas. One way to see the emergence of discovery informatics is to strip away the domain modifiers and recognize the universality of every application area and organization wanting to take advantage of its data.

Discovery informatics techniques are very influential across professions and domains:

- Industry has been using discovery informatics as a fresh approach to design. The discovery and visualization methods from Purdue University helped Caterpillar to design better rubber parts for its heavy equipment products and Lubrizol to improve its fuel additives. "These projects are success stories about how this unique strategy, called 'Discovery Informatics,' has been applied to real-world, product-design problems" (Center for Catalyst Design, 2007).
- In the financial community, the FinCEN Artificial Intelligence System (FAIS) uses discovery informatics methods to help identify money laundering and other crimes. Rule-based inference is used to detect networks of relationships among people and bank accounts so that human analysts can focus their attention on potentially suspicious behavior (Senator, Goldberg, and Wooton, 1995).
- In the education profession, there are exciting scenarios that suggest the potential for discovery informatics techniques. In one example, a knowledge manager works with a special education teacher to evaluate student progress. The data warehouse holds extensive data on students, their performance, teachers, evaluations, and course content. The educators are pleased that the aggregate performance data for the school is satisfactory. However, with discovery informatics capabilities, the story does not need to end there. The data warehouse permits finer granularity examination, and the discovery informatics tools are able to find patterns that are hidden by the summaries. The tools reveal that examinations involving word problems show much poorer performance for certain students. Further analysis shows that this outcome was also true in previous courses for these students. Now the educators have the specific data enabling them to take action, in the form of targeted tutorials for specific students

to address the problem (Tsantis and Castellani, 2001).

- Bioinformatics requires the intensive use of information technology and computer science to address a wide array of challenges (Watkins, 2001). One example illustrates a multi-step computational method to predict gene models, an important activity that has been addressed to date by combinations of gene prediction programs and human experts. The new method is entirely automated, involving optimization algorithms and the careful integration of the results of several gene prediction programs, including evidence from annotation software. Statistical scoring is implemented with decision trees to show clearly the gene prediction results (Allen, Pertea, and Salzberg, 2004).

TRENDS IN DISCOVERY INFORMATICS

There are many indications that discovery informatics will only grow in relevance. Storage costs are dropping; for example, the cost per gigabyte of magnetic disk storage declined by a factor of 500 from 1990 to 2000 (University of California at Berkeley, 2003). Our stockpiles of data are expanding rapidly in every field of endeavor. Businesses at one time were comfortable with operational data summarized over days or even weeks. Increasing automation led to point-of-decision data on transactions. With online purchasing, it is now possible to know the sequence of clickstreams leading up to the sale. So the granularity of the data is becoming finer, as businesses are learning more about their customers and about ways to become more profitable. This business analytics is essential for organizations to be competitive.

A similar process of finer granular data exists in bioinformatics (Dubitsky, et al., 2007). In the human body, there are 23 pairs of human chromosomes, approximately 30,000 genes, and more than one million proteins (Watkins, 2001). The advances in decoding the human genome are remarkable, but it is proteins that "ultimately regulate metabolism and disease in the body" (Watkins, 2001, p. 27). So the challenges for bioinformatics continue to grow along with the data. An indication of the centrality of data mining to research in the biological sciences is the launching in 2007 of the *International Journal of Data Mining and Bioinformatics*.

THE DISCOVERY INFORMATICS COMMUNITY

As discovery informatics has evolved, there has been a steady growth in the establishment of centers, laboratories, and research programs. The College of Charleston launched the first undergraduate degree program in the field in 2005. The interdisciplinary program addresses the need to gain knowledge from large datasets and complex systems, while introducing problem-solving skills and computational tools that span traditional subjects of artificial intelligence, logic, computer science, mathematics, and learning theory.

In addition to the Laboratory for Discovery Informatics at the Johns Hopkins University, laboratories have emerged at Rochester Institute of Technology and Purdue University. Discovery informatics research at Purdue has highlighted the use of knowledge discovery methods for product design. As a further indication of the broad reach of this discovery informatics approach, the Purdue research has been led by professors from the School of Chemical Engineering in their Laboratory for Intelligent Process Systems.

CONCLUSION

Discovery informatics is an emerging methodology that promotes a crosscutting and integrative view. It looks across both technologies and application domains to identify and organize the techniques, tools, and models that improve data-driven discovery.

There are significant research questions as this methodology evolves. Continuing progress will be eagerly received from efforts in individual strategies for knowledge discovery and machine learning, such as the excellent contributions in (Koza et al., 2003). An additional opportunity is to pursue the recognition of unifying aspects of practices now associated with diverse disciplines. While the anticipation of new discoveries is exciting, the evolving practical application of discovery methods needs to respect individual privacy and a diverse collection of laws and regulations. Balancing these requirements constitutes a significant

and persistent challenge as new concerns emerge and laws are drafted.

Because of its essential role in the drug discovery process, discovery informatics is still associated strongly with the pharmaceutical research and industry. However, more recent explorations have confirmed its value in product design and other domains. Community-building around discovery informatics now includes university degree programs and research laboratories with diverse missions. All of these developments provide further evidence that as the challenges and opportunities of the 21st century unfold, discovery informatics is poised to help people and organizations learn as much as possible from the world's abundant and ever growing data assets.

REFERENCES

Agresti, W. W. (2000). Knowledge management. *Advances in Computers 53*, 171-283.

Agresti, W. W. (2003). Discovery informatics. *Communications of the ACM 46 (8)*, 25-28.

Allen, J. E., Pertea, M., & Salzberg, S. L. (2004). Computational gene prediction using multiple sources of evidence. *Genome Research 14*, 142-148.

Bergeron, B. (2003). *Bioinformatics Computing.* Upper Saddle River, NJ: Prentice Hall.

Bramer, M. (2007). *Principles of Data Mining*. London: Springer.

Center for Catalyst Design (2007, October 1). Center for catalyst design: Discovery informatics. The Internet. <https://engineering.purdue.edu/Engr/Research/Focus/2004/

CenterforCatalystDesignDiscoveryInformatics>

Dubitsky, W., Granzow, M., and Berrar, D. P. (Editors) (2007). *Fundamentals of Data Mining in Genomics and Proteomics*. New York: Springer Science+Business Media.

Engelbrecht, A. P. (2007) *Computational Intelligence: An Introduction*. West Sussex, England: John Wiley & Sons.

Heintze, N. (2004, April 2). Scalable document fingerprinting. Carnegie Mellon University. The Internet <http://www2.cs.cmu.edu/afs/cs/user/nch/www/koala/main.html>

Henzinger, M. (2007). Search technologies for the Internet. *Science 27*, 468 – 471.

Houston, A. & Chen, H. (2004, April 2). A path to concept-based information access: From national collaboratories to digital libraries. University of Arizona. The Internet <site:http://ai.bpa.arizona.edu/go/intranet/papers/Book7.pdf>

Koza, J. R., Keane, M. A., Streeter, M. J., Mydlowec, W., Yu, J., and Lanza, G. (Editors) (2003). *Genetic Programming IV: Routine Human-Competitive Machine Intelligence.* Dordrecht, The Netherlands: Kluwer Academic Publishers.

Liebowitz, J. (2006). *Strategic Intelligence: Business Intelligence, Competitive Intelligence, and Knowledge Management*. New York: Auerbach/Taylor & Francis.

Marakas, G. M. (2003). *Modern Data Warehousing, Mining, and Visualization.* Upper Saddle River, NJ: Prentice Hall.

Moran, M. (2008). *Do It Wrong Quickly*. Upper Saddle River, NJ: IBM Press/Pearson.

Prahalad, C. K., & Hamel, G. (1990). The core competence of the corporation. *Harvard Business Review (3)*, 79-91.

Rautiainen M., Ojala T., & Seppänen T. (2004). Analysing the performance of visual, concept and text features in content-based video retrieval. *Proc. 6th ACM SIGMM International Workshop on Multimedia Information Retrieval, New York, NY, 197-205.*

Senator, T. E., Goldberg, H. G., & Wooton, J. (1995). The financial crimes enforcement network AI system (FAIS): Identifying potential money laundering from reports of large cash transactions. *AI Magazine 16*, 21-39.

Smashing Magazine, (2007, October 1) Data visualization: Modern approaches. The Internet.

<http://www.smashingmagazine.com/2007/08/02/data-visualization- modern-approaches>

Taniar, D, (2007). *Data Mining and Knowledge Discovery Technologies*. Hershey, PA:IGI Publishing.

Tsantis, L. & Castellani, J. (2001). Enhancing learning environments through solution-based knowledge discovery tools: Forecasting for self-perpetuating systemic reform. *Journal of Special Education Technology 16*, 39-52.

University of California at Berkeley (2003, June 11). How much information? School of Information Management and Systems. The Internet <http://www.sims.berkeley.edu/research/projects/how-much-info/how-much-info.pdf>

Wang, J. (2003). *Data Mining: Opportunities and Challenges*. Hershey, PA: Idea Group Publishing.

Watkins, K. J. (2001). Bioinformatics. *Chemical & Engineering News 79*, 26-45.

Weinberger, D. (2007). *Everything Is Miscellaneous: The Power of the New Digital Disorder.* New York: Times Books.

KEY TERMS

Clickstream: The sequence of mouse clicks executed by an individual during an online Internet session.

Data Mining: The application of analytical methods and tools to data for the purpose of identifying patterns and relationships such as classification, prediction, estimation, or affinity grouping.

Discovery Informatics: The study and practice of employing the full spectrum of computing and analytical science and technology to the singular pursuit of discovering new information by identifying and validating patterns in data.

Evolutionary Computation: Solution approach guided by biological evolution, which begins with potential solution models, then iteratively applies algorithms to find the fittest models from the set to serve as inputs to the next iteration, ultimately leading to a model that best represents the data.

Knowledge Management: The practice of transforming the intellectual assets of an organization into business value.

Neural Networks: Learning systems, designed by analogy with a simplified model of the neural connections in the brain, which can be trained to find nonlinear relationships in data.

Rule Induction: Process of learning, from cases or instances, if-then rule relationships consisting of an antecedent (if-part, defining the preconditions or coverage of the rule) and a consequent (then-part, stating a classification, prediction, or other expression of a property that holds for cases defined in the antecedent).

Discovery of Protein Interaction Sites

Haiquan Li
The Samuel Roberts Noble Foundation, Inc., USA

Jinyan Li
Nanyang Technological University, Singapore

Xuechun Zhao
The Samuel Roberts Noble Foundation, Inc., USA

INTRODUCTION

Physical interactions between proteins are important for many cellular functions. Since protein-protein interactions are mediated via their interaction sites, identifying these interaction sites can therefore help to discover genome-scale protein interaction map, thereby leading to a better understanding of the organization of living cell. To date, the experimentally solved protein interaction sites constitute only a tiny proportion among the whole population due to the high cost and low-throughput of currently available techniques. Computational methods, including many biological data mining methods, are considered as the major approaches in discovering protein interaction sites in practical applications. This chapter reviews both traditional and recent computational methods such as protein-protein docking and motif discovery, as well as new methods on machine learning approaches, for example, interaction classification, domain-domain interactions, and binding motif pair discovery.

BACKGROUND

Proteins carry out most biological functions within living cells. They interact with each other to regulate cellular processes. Examples of these processes include gene expression, enzymatic reactions, signal transduction, inter-cellular communications and immunoreactions.

Protein-protein interactions are mediated by short sequence of residues among the long stretches of interacting sequences, which are referred to as interaction sites (or binding sites in some contexts). Protein interaction sites have unique features that distinguish them from other residues (amino acids) in protein surface. These interfacial residues are often highly favorable to the counterpart residues so that they can bind together. The favored combinations have been repeatedly applied during evolution (Keskin and Nussinov, 2005), which limits the total number of types of interaction sites. By estimation, about 10,000 types of interaction sites exist in various biological systems (Aloy and Russell, 2004).

To determine the interaction sites, many biotechnological techniques have been applied, such as phage display and site-directed mutagenesis. Despite all these techniques available, the current amount of experimentally determined interaction sites is still very small, less than 10% in total. It should take decades to determine major types of interaction sites using present techniques (Dziembowski and Seraphin, 2004).

Due to the limitation of contemporary experimental techniques, computational methods, especially biological data mining methods play a dominated role in the discovery of protein interaction sites, for example, in the docking-based drug design. Computational methods can be categorized into simulation methods and biological data mining methods. By name, simulation methods use biological, biochemical or biophysical mechanisms to model protein-protein interactions and their interaction sites. They usually take individual proteins as input, as done in protein-protein docking. Recently, data mining methods such as classification and clustering of candidate solutions contributed the accuracy of the approach. Data mining methods learn from large training set of interaction data to induce rules for prediction of the interaction sites. These methods can be further divided into classification methods and pattern mining methods, depending on whether negative data is required. Classification methods require both positive and negative data to develop discriminative features for interaction

sites. In comparison, pattern mining methods learn from a set of related proteins or interactions for over-presented patterns, as negative data are not always available or accurate. Many homologous methods and binding motif pair discovery fall into this category.

MAIN FOCUS

Simulation Methods: Protein-Protein Docking

Protein-protein docking, as a typical simulation method, takes individual tertiary protein structures as input and predicts their associated protein complexes, through simulating the confirmation change such as side-chain and backbone movement in the contact surfaces when proteins are associated into protein complexes. Most docking methods assume that, conformation change terminates at the state of minimal free energy, where free energy is defined by factors such as shape complementarity, electrostatic complementarity and hydrophobic complementarity.

Protein-protein docking is a process of search for global minimal free energy, which is a highly challenging computational task due to the huge search space caused by various flexibilities. This search consists of four steps. In the first step, one protein is fixed and the other is superimposed into the fixed one to locate the best docking position, including translation and rotation. Grid-body strategy is often used at this step, without scaling and distorting any part of the proteins. To reduce the huge search space in this step, various search techniques are used such as, Fast Fourier transformation, Pseudo-Brownian dynamics and molecular dynamics (Mendez et al., 2005). In the second step, the flexibility of side chains is considered. The backbone flexibility is also considered using techniques such as principal component analysis in some algorithms (Bonvin, 2006). Consequently, a set of solutions with different local minima is generated after the first two steps. These solutions are clustered in the third step and representatives are selected (Lorenzen & Zhang, 2007). In the fourth step, re-evaluation is carried out to improve the ranks for nearly native solutions, since the nearly native solutions may not have the best free energy scores due to the flaws of score functions and search algorithms. Supervised data mining techniques have been applied in this step to select the near-native solution, using the accumulative confirmation data for benchmark protein complexes (Bordner and Gorin 2007). Note that in all steps, biological information may be integrated to aid the search process, such as binding sites data (Carter et al., 2005). In the interaction site determination problem, without the guidance of binding sites in docking, the top-ranked interfaces in the final step correspond to the predicted interaction sites. With the guidance of binding sites, the docking algorithms may not contribute remarkably to the prediction of interaction sites since the above steps may be dominated by the guided binding sites.

Although protein-protein docking is the major approach to predict protein interaction sites, the current number of experimentally determined protein structures is much less than that of protein sequences. Even using putative structures, ~ 40% proteins will be failed in protein structure prediction (Aloy et al., 2005), especially for transmembrane proteins. This leaves a critical gap in the protein-protein docking approach.

Classification Methods

Classification methods assume that the features, either in protein sequence or in protein spatial patches, distinguish positive protein interactions from negative non-interactions. Therefore, the distinguishing features correspond to protein interaction sites. The assumption generally holds true but not always.

The first issue in protein interaction classification is to encode protein sequences or structures into features. At least two encoding methods are available. One transforms continuous residues and their associated physicochemical properties in the primary sequence into features (Yan et al., 2004). The other encodes a central residue and its spatially nearest neighbors one time, which is so called spatial patches (Fariselli et al., 2002). The latter encoding is more accurate than the first one because protein structures are more related to interaction sites.

After encoding the features, traditional classification methods such as support vector machine (SVM) and neural networks can be applied to predict interaction sites (Bock & Gough, 2001; Ofran & Rost, 2003). Recently, a two-stage method was proposed (Yan et al., 2004). In the learning phase, both SVM and Bayesian networks produce a model for the continuously encoded residues. In the prediction phase, the SVM model is first applied to predict a class value for each residue, then the Bayesian

model is applied to predict the final class value based on predicted values in SVM model, exploiting the fact that interfacial residues tend to form clusters.

Although classification methods have many advantages, for example, they are good at handling transient complexes which are tough in docking, they have several disadvantages. First, they suffer from unavailability and low quality of the negative data. Second, many classification algorithms apply fixed-length windows in coding, which conflicts with the basic fact that many interaction sites have variable length. Finally, the complicated coding often results in incomprehensibility to the interaction sites from a biological point of view.

Pattern Mining Methods

Pattern mining methods assume that interaction sites are highly conserved in protein homologous data and protein-protein interactions. These conserved patterns about interaction sites are quite different from random expectation and thus, can be revealed even in the absence of negative data. Typical patterns include binding motifs, domain-domain interaction pairs and binding motif pairs.

Binding Motifs from Homologous Proteins

Given a group of homologous proteins, the inherited patterns can be recovered by searching the locally over-represented patterns (so-called motifs) among their sequences, which is often referred to as motif discovery. It is a NP-hard problem and similar to sequential pattern mining but more complicated since its score function is often implicit. The majority of methods discover motifs from primary sequences and can be roughly categorized into pattern-driven, sequence-driven and the combined ones. Pattern-driven methods enumerate all possible motifs with a specific format and output the ones with enough occurrences. For instance, MOTIF (Smith et al., 1990) searches all frequent motifs with three fixed positions and two constant spacings. Sequence-driven methods restrict the candidate patterns to occur at least some times in the group of sequences, for instance, the widely used CLUSTALW (Thompson et al., 1994). Combined methods integrate the strength of pattern-driven and sequence-driven methods. They start from patterns at short lengths and extend them based on conservation of their neighbors in the sequences, for instance, PROTOMAT (Jonassen, 1997). Other motif discovery approaches have also been studied, for example, statistical models such as Hidden Markov models (HMM) and expectation maximization (EM) models.

Motifs can also be discovered from homologous protein structures, which are called structural motifs. Structural motif discovery has been studied from various perspectives, such as frequent common substructure mining (Yan et al., 2005) and multiple structure alignment (Lupyan et al., 2005), impelling by the rapid growth of solved protein structures in recent years.

In general, motif discovery only identifies individual motifs without specifying their interacting partners and thus, can't be considered as complete interaction sites. To reveal both sides of interaction sites, individual motifs can be randomly paired and their correlation can be evaluated by protein-protein interaction data, as done by Wang et al. (2005). Even though, the discovered motif pairs can not guarantee to be interaction sites or binding sites. The rationale is that binding and folding are often interrelated and they could not be distinguished only from homologous proteins. Since homologous proteins share more folding regions than bindings regions, the discovered motifs by sequence or structure conservation are more likely to be folding motifs rather than binding motifs (Kumar et al., 2000). To identify the complete interaction sites, protein interaction information should be taken into consideration in the early stage of learning.

Domain-Domain Pairs from Protein-Protein Interactions

Domain-domain interactions, which are closely related to protein interaction sites, have been widely studied in recent years, due to the well-acknowledged concept of domain and the abundantly available data about protein interactions. Many domains contain regions for interactions and involve in some biological functions.

From a data mining perspective, each protein sequence is a sequence of domains and the target patterns are correlated domain pairs. The correlated pairs have been inferred by various approaches. Sprinzak and Margalit (2001) extracted all over-represented domain pairs in protein interaction data and initially termed them as correlated sequence-signatures. Wojcik and Schachter (2001) generated interacting domain pairs from protein cluster pairs with enough interactions, where the protein clusters are formed by proteins with enough sequence similarities and common interacting partners. Deng et

al. (2002) used maximum-likelihood to infer interacting domain pairs from a protein interaction dataset, by modeling domain pairs as random variables and protein interactions as events. Ng et al. (2003) inferred domain pairs with enough integrated scores, integrating evidences from multiple interaction sources.

Although domain-domain interactions imply abundant information about interaction sites, domains are usually very lengthy, in which only small parts involve binding while most regions contribute to folding. On the contrary, some interaction sites may not occur in any domain. Therefore, the study of domain-domain interactions is not enough to reveal interaction sites, although helpful.

Binding Motif Pairs from Protein-Protein Interactions

To fill the gap left by all above techniques, a novel concept of binding motif pairs has been proposed to define and capture more refined patterns at protein interaction sites hidden among abundant protein interaction data (Li and Li, 2005a). Each binding motif pair consists of two traditional protein motifs, which are usually more specific than domains. Two major methods were developed for the discovery of binding motif pairs.

The first method is based on a fixed-point theorem, which describes the stability under the resistance to some transformation at some special points; that is, the points remain unchanged by a transformation function. Here the stability corresponds to the biochemical laws exhibited in protein-protein interactions. The points of function are defined as protein motif pairs. This transformation function is based on the concept of occurrence and consensus. The discovery of fixed points, or the stable motif pairs, of the function is an iterative process, undergoing a chain of changing but converging patterns (Li and Li, 2005b).

The selection of the starting points for this function is difficult. An experimentally determined protein complex dataset was used to help in identifying meaningful starting points so that the biological evidence is enhanced and the computational complexity is greatly reduced. The consequent stable motif pairs are evaluated for statistical significance, using the unexpected frequency of occurrence of the motif pairs in the interaction sequence dataset. The final stable and significant motif pairs are the binding motif pairs finally targeted (Li and Li, 2005a).

The second method is based on the observation of frequently occurring substructures in protein interaction networks, called interacting protein-group pairs corresponding to maximal complete bipartite subgraphs in the graph theory. The properties of such substructures reveal a common binding mechanism between the two protein sets attributed to all-versus-all interaction between the two sets. The problem of mining interacting protein groups can be transformed into the classical problem of mining closed patterns in data mining (Li et al., 2007). Since motifs can be derived from the sequences of a protein group by standard motif discovery algorithms, a motif pair can be easily formed from an interacting protein group pair (Li et al., 2006).

FUTURE TRENDS

Current approaches to discover interaction sites from protein-protein interactions usually suffer from ineffective models, inefficient algorithms and lack of sufficient data. Many studies can be done in the future exploring various directions of the problem. For example, new machine learning methods may be developed to make full use of existing knowledge on the successfully docked protein complexes. More effective models may be proposed to discover binding motifs from homologous proteins or from domain-domain interacting pairs, through effective distinguishing of binding patterns from folding patterns. Other diverse models such as quasi-bipartite may be developed to discover binding motif pairs. Overall, the focus of studying interactions will then move from protein-protein interactions and domain-domain interactions to motif-motif interactions.

CONCLUSION

In this mini review, we have discussed various methods for the discovery of protein interaction sites. Protein-protein docking is the dominant method but it is constrained by the limited amount of protein structures. Classification methods constitute traditional machine learning methods but suffer from inaccurate negative data. Pattern mining methods are still incapable to distinguish binding patterns from folding patterns, except a few work based on binding motif pairs. Overall, current biological data mining methods are far from perfect. Therefore, it is valuable to develop novel methods in

the near future to improve the coverage rate, specificity and accuracy, especially by using the fast growing protein-protein interaction data, which are closely related to the interaction sites.

REFERENCES

Aloy, P., Pichaud, M., & Russell, R. (2005). Protein complexes: structure prediction challenges for the 21st century. *Current Opinion in Structural Biology*, 15(1), 15-22.

Aloy, P., & Russell, R. (2004). Ten thousand interactions for the molecular biologist. *Nature Biotechnology, 22*(10), 1317-1321.

Agrawal, R & Srikant, R. (1995). Mining sequential Patterns. *Proceedings of International Conference on Data Engineering (ICDE)* (pp. 3-14).

Bonvin, A. (2006). Flexible protein-protein docking. *Current Opinion in Structural Biology, 16*(2), 194-200.

Bordner. AJ & Gorin, AA (2007). Protein docking using surface matching and supervised machine learning. *Proteins*, 68(2), 488-502.

Carter, P., Lesk, V., Islam, S., & Sternberg, M. (2005). Protein-protein docking using 3d-dock in rounds 3,4, and 5 of CAPRI. *Proteins, 60*(2), 281-288.

Deng, M., Mehta, S., Sun, F., & Chen, T. (2002). Inferring domain-domain interactions from protein-protein interactions. *Genome Research, 12*(10), 1540-1548.

Dziembowski, A., & Seraphin, B. (2004). Recent developments in the analysis of protein complexes. *FEBS Letters, 556*(1-3), 1-6.

Fariselli, P., Pazos, F., Valencia, A. & Casadio, R. (2002). Prediction of protein-protein interaction sites in heterocomplexes with neural networks. *European Journal of Biochemistry, 269*(5), 1356-1361.

Bock, J.R., & Gough, D.A. (2001). Predicting protein-protein interactions from primary structure. *Bioinformatics, 17*(5), 455-460.

Jonassen, I. (1997). Efficient discovery of conserved patterns using a pattern graph. *Computer Applications in Biosciences, 13*(5), 509-522.

Keskin, O., & Nussinov, R. (2005). Favorable scaffolds: proteins with different sequence, structure and function may associate in similar ways. *Protein Engineering Design & Selection, 18*(1), 11-24.

Kumar, S., Ma, B., Tsai, C., Sinha, N., & Nussinov, R. (2000). Folding and binding cascades: dynamic landscapes and population shifts. *Protein Science, 9*(1), 10-19.

Li, H., & Li, J. (2005a). Discovery of stable and significant binding motif pairs from PDB complexes and protein interaction datasets. *Bioinformatics, 21*(3), 314-324.

Li, H., Li, J., & Wong, L. (2006). Discovering motif pairs at interaction sites from protein sequences on a proteome-wide scale. *Bioinformatics, 22*(8), 989-996.

Li, J., & Li, H. (2005b). Using fixed point theorems to model the binding in protein-protein interactions. *IEEE transactions on Knowledge and Data Engineering, 17*(8), 1079-1087.

Li, J., Liu, G. Li, H., & Wong, L. (2007). A correspondence between maximal biclique subgraphs and closed pattern pairs of the adjacency matrix: a one-to-one correspondence and mining algorithms. *IEEE transactions on Knowledge and Data Engineering*, 19(12), 1625-1637.

Lorenzen, S & Zhang, Y (2007). Identification of near-native structures by clustering protein docking conformations. *Proteins*, 68(1), 187-194.

Lupyan, D., Leo-Macias, A., & Ortiz, A. (2005). A new progressive-iterative algorithm for multiple structure alignment. *Bioinformatics, 21*(15), 3255-3263.

Mendez, R., Leplae, R., Lensink, M., & Wodak, S. (2005). Assessment of CAPRI predictions in rounds 3-5 shows progress in docking procedures. *Proteins, 60*(2), 150-169.

Ng, S., Zhang, Z., & Tan, S. (2003). Integrative approach for computationally inferring protein domain interactions. *Bioinformatics, 19*(8), 923-929.

Ofran, Y., & Rost, B. (2003). Predicted protein-protein interaction sites from local sequence information. *FEBS Letters, 544*(3), 236-239.

Smith, H., Annau, T. M., & Chandrasegaran, S. (1990). Finding sequence motifs in groups of functionally related proteins. *Proceedings of National Academy of Sciences, 87*(2), 826-830.

Sprinzak, E., & Margalit, H. (2001). Correlated sequence-signatures as markers of protein-protein interaction. *Journal of Molecular Biology, 311*(4), 681-692.

Thompson, J., Higgins, D., & Gibson, T. (1994). ClustalW: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research, 22*(22), 4673-4680.

Wang, H., Segal, E., Ben-Hur, A., Koller, D., & Brutlag, D. (2005). Identifying protein-protein interaction sites on a genome-wide scale. *Advances in Neural Information Processing Systems 17* (pp. 1465-1472). USA.

Wojcik, J., & Schachter, V. (2001). Protein-protein interaction map inference using interacting domain profile pairs. *Bioinformatics, 17*(Suppl 1), S296-S305.

Yan, C., Dobbs, D., & Honavar, V. (2004). A two-stage classifier for identification of protein-protein interface residues. *Bioinformatics, 20*(Suppl 1), I371-I378.

Yan, X., Yu, P. S., & Han, J. (2005). Substructure similarity search in graph databases. In *proceedings of 2005 ACM-SIGMOD International Conference on Management of Data* (pp. 766-777). Baltimore, Maryland.

KEY TERMS

Binding Motif Pairs: A pair of binding motifs which interact with each other to determine a type of protein interaction.

Binding Motifs: The patterns which describe a group of sequences or structures that bind or interact with a specific target.

Bioinformatics: The research, development, or application of computational tools and approaches for expanding the use of biological, medical, behavioral, or health data, including those used to acquire, store, organize, analyze, or visualize such data.

Domain-Domain Interactions: The binding or association among two or more domains due to their sequence or structure preference.

Protein Interaction Sites: The regions of proteins associated with the other interacting partner during protein-protein interactions.

Protein-Protein Docking: The determination of the molecular structures of complexes formed by two or more proteins without the need for experimental measurement.

Protein-Protein Interactions: The association of protein molecules and the study of these associations from the perspective of biochemistry, signal transduction and networks.

D

Distance-Based Methods for Association Rule Mining

Vladimír Bartík
Brno University of Technology, Czech Republic

Jaroslav Zendulka
Brno University of Technology, Czech Republic

INTRODUCTION

Association rules are one of the most frequently used types of knowledge discovered from databases. The problem of discovering association rules was first introduced in (Agrawal, Imielinski & Swami, 1993). Here, association rules are discovered from transactional databases –a set of transactions where a transaction is a set of items. An association rule is an expression of a form $A \Rightarrow B$ where A and B are sets of items. A typical application is market basket analysis. Here, the transaction is the content of a basket and items are products. For example, if a rule *milk* $\wedge$ *juice* $\Rightarrow$ *coffee* is discovered, it is interpreted as: "If the customer buys milk and juice, s/he is likely to buy coffee too." These rules are called *single-dimensional Boolean association rules* (Han & Kamber, 2001). The potential usefulness of the rule is expressed by means of two metrics – support and confidence.

A lot of algorithms have been developed for mining association rules in transactional databases. The best known is the Apriori algorithm (Agrawal & Srikant, 1994), which has many modifications, e.g. (Kotásek & Zendulka, 2000). These algorithms usually consist of two phases: discovery of frequent itemsets and generation of association rules from them. A frequent itemset is a set of items having support greater than a threshold called minimum support. Association rule generation is controlled by another threshold referred to as minimum confidence.

Association rules discovered can have a more general form and their mining is more complex than mining rules from transactional databases. In relational databases, association rules are ordinarily discovered from data of one table (it can be the result of joining several other tables). The table can have many columns (attributes) defined on domains of different types. It is useful to distinguish two types of attributes.

A *categorical attribute* (also called nominal) has a finite number of possible values with no ordering among the values (e.g. a country of a customer).

A *quantitative attribute* is a numeric attribute, domain of which is infinite or very large. In addition, it has an implicit ordering among values (e.g. age and salary of a customer).

An association rule (*Age* = [20…30]) $\wedge$ (*Country* = "Czech Rep.") $\Rightarrow$ (Salary = [1000$...2000$]) says that if the customer is between 20 and 30 and is from the Czech Republic, s/he is likely to earn between 1000$ and 2000$ per month. Such rules with two or more predicates (items) containing different attributes are also called *multidimensional association rules*. If some attributes of rules are quantitative, the rules are called quantitative association rules (Han & Kamber, 2001).

If a table contains only categorical attributes, it is possible to use modified algorithms for mining association rules in transactional databases. The crucial problem is to process quantitative attributes because their domains are very large and these algorithms cannot be used. Quantitative attributes must be *discretized*.

This article deals with mining multidimensional association rules from relational databases, with main focus on distance-based methods. One of them is a novel method developed by the authors.

BACKGROUND

There are three basic approaches regarding the treatment of quantitative attributes (Han & Kamber, 2001). First one uses a predefined set of ranges (or, in general, a concept hierarchy) to replace the original numeric values of a quantitative attribute by ranges that represent intervals of values. This discretization occurs prior to applying a mining algorithm. It is *static* and

predetermined. In the second approach, quantitative attributes are initially discretized statically. The resulting ranges are then combined during the mining algorithm. Therefore, the discretization process is *dynamic*. The third approach tries to define ranges based on semantic meaning of the data. This discretization is dynamic too. It considers distance between data points. Discretization and mining methods based on this approach are referred to as *distance-based methods*.

There are two basic methods of *static discretization*: equi-depth and equi-width discretization. Equi-depth discretization lies in creating intervals that contain the same number of values. Equi-width discretization creates ranges of the same size. These methods are very simple, but the result of discretization can be unsuitable because some important associations may be lost. This is caused by the fact that two very near values can be in two different ranges. Sometimes one cluster of values, which might be represented by one predicate in an association rule, is divided into two ranges.

The following two methods are representatives of the second approach mentioned above.

A method for mining quantitative association rules was proposed in (Srikant & Agrawal, 1996). The number of intervals, which will be created, is determined by means of a measure referred to as K-partial completeness, which guarantees the acceptable loss of information. This measure is represented by a number K, which is higher than 1. This value is used to determine number of intervals N:

$$N = \frac{2}{minsup \cdot (K-1)} \qquad (1)$$

where *minsup* is a minimum support threshold.

After initial equi-depth discretization, neighboring intervals are joined together to form intervals having sufficient support. These intervals are then used to discover frequent itemsets.

Zhang extended the method to *fuzzy quantitative association rules* in (Zhang, 1999). Here, association rules can contain fuzzy terms too. In the phase of discretization, creating of intervals is combined with creating of fuzzy terms over quantitative attributes.

A method for mining *optimized association rules* proposed in (Fukuda, Morimoto, Morishita & Tokuyama, 1996) uses no additional measure for the interestingness of an association rule. Because the minimum support and confidence thresholds are antipodal, either rules referred to as optimized support rules or optimized confidence rules are obtained with this method. The goal of the optimized support rules is to find rules with support as high as possible that satisfy minimum confidence. Similarly, the optimized confidence rules maximize confidence of the rule while satisfying the minimum support requirement. The method uses the initial equi-depth discretization of quantitative attributes. Then the method continues with finding optimal interval for a given form of an association rule. The method discovers association rules referred to as constraint-based association rules where the form of rules to be mined must be defined, for example: $(Age \in [v_1, v_2]) \wedge (Country = X) \Rightarrow (Salary \in [v_3, v_4])$. Another method for mining optimized association rules is proposed in (Xiaoyong, Zhibin & Naohiro, 1999).

MAIN FOCUS

Distance-based methods try to respect semantics of data in such a way that the discretization of quantitative attributes reflects the distances between numeric values. The intervals of numeric values are constructed dynamically to contain clusters of values lying close to each other. The main objective of distance-based methods is to minimize loss of information caused by discretization.

The distance-based approach can either be applied as a discretization method or as part and parcel of a mining algorithm.

The first distance-based method was proposed in (Miller & Yang, 1997). Here, the clustering methods are used to create intervals. This algorithm works in two phases; in the first one, interesting clusters over quantitative attributes are found (e.g., with clustering algorithm Birch (Zhang, Ramakrishnan & Livny, 1996) and these clusters are used to generate frequent itemsets of clusters and association rules containing them.

The result of the process is a set of association rules of a form $C_{X1} \wedge \ldots \wedge C_{Xn} \Rightarrow C_{Y1} \wedge \ldots \wedge C_{Yn}$, where C_X and C_Y are clusters over quantitative attributes X and Y. Xi and Yj are pairwise disjoint sets of quantitative attributes. Except minimum support and confidence, an association rule must meet the condition of an *association degree*. To determine the value of the association degree of a rule $C_{Xi} \Rightarrow C_{Yi}$, we need to know if values of the attribute X in the cluster C_{Yi} are inside the cluster C_{Xi}

or close to it. To verify it, we can compute the average value of the attribute X inside the cluster C_{Xi} and inside the cluster C_{Yi} and compare these two values.

The problem of this method appears if there are several quantitative attributes, because they often have very different ranges. We need to normalize them to be comparable. Normalization must be performed before clustering. There are a lot of statistical normalization methods but their use depends on a detailed understanding of data and relationships between them (Milligan, 1995). This method is suitable for mining association rules in a table with only quantitative attributes.

An Adaptive Method for Numerical Attribute Merging was proposed in (Li, Shen & Topor, 1999). The method combines the approach of merging small intervals into larger ones with clustering. At first, each quantitative value is assigned to its own interval (containing only one value). Then, two neighboring intervals with lowest difference are merged together. The criterion that controls the merging process respects both distance and density of quantitative values.

Assume that for each interval, a representative center is defined. The intra-interval distance is defined as the average of distances of the representative center and each individual value inside the interval. The difference of two neighboring intervals can be determined as the intra-interval distance of an interval that is created as a union of them. In every iteration, neighboring intervals with the lowest difference are merged. The process is stopped, if all differences of intervals are higher than a value of $3d$, where d is the average distance of adjacent values of the quantitative attribute being processed.

Another method of interval merging described in (Wang, Tay & Liu, 1998) uses an interestingness measure. The idea is to merge intervals only if the loss of information caused by merging is justified by improved interestingness of association rules. The method uses several interestingness measures. Most of them are based on support and confidence of a resultant association rule.

A method which uses evolutionary algorithm to find optimal intervals was proposed in (Mata, Alvarez & Riquelme, 2002). An evolutionary algorithm is used to find the most suitable size of the intervals that conforms an itemset. The objective is to get high support value without being the intervals too wide. Discretization is performed during the mining of frequent itemsets.

At first, the initial population of frequent itemsets is obtained. Then, the population is improved in each iteration. The quality of discretization is determined by means of a fitness function. The fitness function reflects the width of intervals over quantitative attributes, support and the similarity with another intervals used in other frequent itemsets.

Unfortunately, the methods described above can produce a lot of uninteresting rules that must be eliminated. To cope with this problem, several post-processing methods have been developed to reduce the number of association rules (Li, Shen & Topor, 1999; Gupta, Strehl & Gosh, 1999).

In addition, distance-based methods described above, have some disadvantages. If a relational table contains both categorical and quantitative attributes, the ability to generate association rules containing both categorical and quantitative values is often small.

In next paragraphs, a method called *Average Distance Based Method* (Bartík & Zendulka, 2003), which was designed for mining association rules in a table that can contain both categorical and quantitative attributes, is described.

The method can be characterized by the following important features:

- Categorical and quantitative attributes are processed separately each other.
- Quantitative attributes are discretized dynamically.
- The discretization is controlled by means of two thresholds called maximum average distance and precision. The values of the thresholds are specified for each quantitative attribute.

Assume a quantitative attribute A and its value v. Let I be an interval of N values (data points) v_i ($1 \le i \le N$) of A such that $v \in I$. The *average distance* from v in I is defined as:

$$AD_I(v) = \sum_{i=1}^{N} \frac{(v - v_i)}{N}, \qquad (2)$$

The *maximum average distance* ($maxAD_A$) is a threshold for the average distance measured on values of an attribute A.

A value v of a quantitative attribute A is called an *interesting value* if there is an interval I that includes v with the following two properties:

- It can extend an existing frequent itemset by one item, i.e. the extended itemset satisfies minimum support.
- It holds $AD_I(v) \leq maxAD_A$.

The *precision* P_A is a threshold that determines the granularity of searching for interesting values of an attribute *A*. It is the minimum distance between two successive interesting values of *A*.

If *A* is a quantitative attribute, predicates of association rules containing A have form $(A, = v, maxAD_A)$.

Similarly to many other algorithms, mining association rules is performed in two steps:

1. Finding frequent itemsets, i.e. itemsets satisfying the minimum support threshold *minsup*.
2. Generating strong association rules, i.e. rules that satisfy the minimum confidence threshold *minconf*.

The second step is basically the same as in other well-known algorithms for mining association rules, for example (Agrawal & Srikant, 1994). Therefore, we focus here on the first step only. It can be described in pseudocode as follows:

Algorithm: Average Distance Based Method. Step 1 – finding frequent itemsets.
Input: Table *T* of a relational database; minimum support *minsup*; maximum average distance $maxAD_A$ and precision P_A, for each quantitative attribute of *T*.
Output: *FI*, frequent itemsets in *T*.

```
1.    // process categorical attributes
FI_0 = find_frequent_categorical_itemsets (T, minsup)
2     k = 0
3     // process quantitative attributes one by one
for each quantitative attribute A in T {
3.1     for each frequent itemset fi in FI_k {
3.1.1       V = find_interesting_values (A, minsup,
      maxAD_A, P_A)
3.1.2     for each interesting value v in V {
3.1.2.1       FI_{k+1} = FI_{k+1} ∪ (fi ∪ (A = v, maxAD_A))
          }
        }
      k = k+1
    }
  FI = ⋃_k FI_k
```

First, only categorical attributes are considered and frequent itemsets (referred to as categorical) are found. Any algorithm for mining association rules in transactional data can be used. Quantitative attributes are iteratively processed one by one. For each frequent itemset *fi* the quantitative attribute being processed is dynamically discretized. It means that interesting values of the attribute are found using the average distance measure and the maximum average distance and the precision thresholds.

Assume that a quantitative attribute *A* is being processed. Each frequent itemset discovered in the previous iteration determines the values of *A* that must be taken into account in searching for interesting values of *A*. Each candidate value *v* is determined by means of precision P_A. To satisfy the minimum support threshold *minsup*, the number *N* of considered neighboring values of *v* in an interval *I* used in computation of the average distance (2) is

$$N = minsup * numrows, \qquad (3)$$

where *numrows* is the total number of rows in the mined table. If $AD_I(v) \leq maxAD_A$ then the value *v* is interesting and the frequent itemset can be extended by a new item $(A = v, maxAD_A)$.

The discovered frequent itemsets are dependent on the order, in which quantitative attributes are processed. Therefore, a heuristics for the order that tries to maximize the number of discovered frequent itemsets was designed. For each quantitative attribute *A* in the table, the following function is computed:

$$H(A) = \frac{\max AD_A}{P_A} \cdot (1 - \frac{mis\sin g}{numrows}), \qquad (4)$$

where missing is the number of rows in which the value of the attribute A is missing. The attributes should be processed in descending order of the H(A) values.

FUTURE TRENDS

Methods and algorithms for mining association rules in relational databases are constantly the subject of research. The approach which is often applied – static discretization of quantitative attributes, relies heavily on the experience of the user. Distance-based methods try to discretize quantitative attributes dynamically,

which can increase the chance of discovery all the most important association rules.

There is another challenge for distance-based methods. Nowadays, complex data as object oriented, multimedia data, text and XML documents are constantly used and stored in databases more frequently. Distance-based methods can be adapted, in some cases, for mining such data.

CONCLUSION

The aim of this article was to provide a brief overview of methods for mining multidimensional association rules from relational databases. It was focused mainly on distance-based methods. In addition, a novel method called Average distance based method has been presented. It can be considered as a framework for combination of algorithms for mining association rules over categorical attributes and a new method of dynamic discretization of quantitative attributes. This framework allows using any method for processing of categorical attributes. In addition, it makes it possible to achieve better ability to generate association rules containing both categorical and quantitative attributes (Bartík & Zendulka, 2003).

REFERENCES

Agrawal, R., Imielinski, T., & Swami, A. (1993). Mining Association Rules Between Sets of Items in Large Databases. *Proceedings of the ACM SIGMOD International Conference on Management of Data* (pp. 207-216), Washington, USA.

Agrawal, R., & Srikant, R. (1994). Fast Algorithms for Mining Association Rules. *Proceedings of the 20th VLDB Conference* (pp. 487-499), Santiago de Chile, Chile.

Bartík, V., & Zendulka J. (2003). Mining Association Rules from Relational Data - Average Distance Based Method, *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA and ODBASE* (pp. 757-766), Catania, Italy.

Fukuda, T., Morimoto, Y., Morishita, S., & Tokuyama, T. (1996). Mining Optimized Association Rules For Numeric Attributes. *Proceedings of ACM PODS'96* (pp. 182-191), Montreal, Canada.

Gupta, G., Strehl, A., & Ghosh, J. (1999). Distance Based Clustering of Association Rules, *Proceedings of ANNIE 1999* (pp. 759-764), New York, USA.

Han, J., Kamber, M. (2001). Data Mining – Concepts And Techniques, San Francisco, USA: Morgan Kaufmann Publishers.

Kotásek, P., Zendulka, J. (2000). Comparison of Three Mining Algorithms For Association Rules, *Proceeding of MOSIS 2000 – Information Systems Modeling ISM 2000* (pp. 85-90), Rožnov pod Radhoštěm, Czech Republic.

Li, J., Shen, H., & Topor, R. (1999). An Adaptive Method of Numerical Attribute Merging for Quantitative Association Rule Mining, *Proceedings of the 5th international computer science conference (ICSC)* (pp. 41-50), Hong Kong, China.

Mata, J., Alvarez, J. L., & Riquelme, J. C. (2002). Discovering Numeric Association Rules via Evolutionary Algorithm. *Proceedings of PAKDD 2002, Lecture Notes In Artificial Intelligence, Vol. 2336* (pp. 40-51), Taipei, Taiwan.

Miller, R. J., & Yang, Y. (1997). Association Rules over Interval Data. *Proceedings of 1997 ACM SIGMOD* (pp. 452-461), Tucson, Arizona, USA.

Milligan, G. W. (1995). Clustering Validation: Results and Implications for Applied Analyses. Clustering and Classification (pp. 345-375), World Scientific Publishing, River Edge, New Jersey.

Srikant, R., & Agrawal, R. (1996). Mining Quantitative Association Rules In Large Relational Tables. *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data* (pp. 1-12), Montreal, Canada.

Wang, K., Tay S. H. W., & Liu, B. (1998). Interestingness-Based Interval Merger for Numeric Association Rules. *Proceedings of 4th International Conference Knowledge Discovery and Data Mining (KDD)* (pp. 121-128), New York, USA.

Xiaoyoung, D., Zhibin, L., & Naohiro, I. (1999). Mining Association Rules on Related Numeric Attributes. *Proceedings of PAKDD 99, Lecture Notes In Artificial Intelligence, Vol. 1574* (s. 44-53), Beijing, China.

Zhang, T., Ramakrishnan, R., Livny, M. (1996). Birch: An Efficient Data Clustering Method for Very Large Databases. *Proceedings of the ACM SIGMOD International Conference on Management of Data* (pp. 103-114), Montreal, Canada.

Zhang, W. (1999). Mining Fuzzy Quantitative Association Rules. *Proceedings of 11th IEEE International Conference on Tools with Artificial Intelligence* (pp. 99-102), Chicago, USA.

KEY TERMS

Association Rule: For a database of transactions containing some items, it is an implication of the form $A \Rightarrow B$, where A and B are sets of items. The semantics of the rule is: If a transaction contains items A, it is likely to contain items B. It is also referred to as a single-dimensional association rule.

Average Distance Based Method: A method for mining quantitative association rules. It separates categorical and quantitative attributes processing, and employs dynamic discretization controlled by an average distance measure and maximum average distance and precision thresholds.

Categorical Attribute: An attribute that has a finite number of possible values with no ordering among the values (e.g. a country of a customer)

Discretization: In data mining, it is the process of transferring quantitative attributes into categorical ones by using intervals instead of individual numeric values.

Distance-Based Method: A method for mining association rules in which the discretization of quantitative attributes is performed dynamically, based on the distance between numeric values.

Dynamic Discretization: Discretization that respects the semantics of data. It is usually performed during the run of a mining algorithm..

Multidimensional Association Rule: An association rule with two or more predicates (items) containing different attributes.

Quantitative Association Rule: A synonym for multidimensional association rules containing at least one quantitative attribute.

Quantitative Attribute: A numeric attribute, domain of which is infinite or very large. In addition, it has an implicit ordering among values (e.g. age and salary).

Static Discretization: Discretization based on a predefined set of ranges or conceptual hierarchy.

Distributed Association Rule Mining

Mafruz Zaman Ashrafi
Monash University, Australia

David Taniar
Monash University, Australia

Kate A. Smith
Monash University, Australia

INTRODUCTION

Data mining is an iterative and interactive process that explores and analyzes voluminous digital data to discover valid, novel, and meaningful patterns (Mohammed, 1999). Since digital data may have terabytes of records, data mining techniques aim to find patterns using computationally efficient techniques. It is related to a subarea of statistics called exploratory data analysis. During the past decade, data mining techniques have been used in various business, government, and scientific applications.

Association rule mining (Agrawal, Imielinsky & Sawmi, 1993) is one of the most studied fields in the data-mining domain. The key strength of association mining is completeness. It has the ability to discover all associations within a given dataset. Two important constraints of association rule mining are support and confidence (Agrawal & Srikant, 1994). These constraints are used to measure the interestingness of a rule. The motivation of association rule mining comes from market-basket analysis that aims to discover customer purchase behavior. However, its applications are not limited only to market-basket analysis; rather, they are used in other applications, such as network intrusion detection, credit card fraud detection, and so forth.

The widespread use of computers and the advances in network technologies have enabled modern organizations to distribute their computing resources among different sites. Various business applications used by such organizations normally store their day-to-day data in each respective site. Data of such organizations increases in size everyday. Discovering useful patterns from such organizations using a centralized data mining approach is not always feasible, because merging datasets from different sites into a centralized site incurs large network communication costs (Ashrafi, David & Kate, 2004). Furthermore, data from these organizations are not only distributed over various locations, but are also fragmented vertically. Therefore, it becomes more difficult, if not impossible, to combine them in a central location. Therefore, Distributed Association Rule Mining (DARM) emerges as an active subarea of data-mining research.

Consider the following example. A supermarket may have several data centers spread over various regions across the country. Each of these centers may have gigabytes of data. In order to find customer purchase behavior from these datasets, one can employ an association rule mining algorithm in one of the regional data centers. However, employing a mining algorithm to a particular data center will not allow us to obtain all the potential patterns, because customer purchase patterns of one region will vary from the others. So, in order to achieve all potential patterns, we rely on some kind of distributed association rule mining algorithm, which can incorporate all data centers.

Distributed systems, by nature, require communication. Since distributed association rule mining algorithms generate rules from different datasets spread over various geographical sites, they consequently require external communications in every step of the process (Ashrafi, David & Kate, 2004; Assaf & Ron, 2002; Cheung, Ng, Fu & Fu, 1996). As a result, DARM algorithms aim to reduce communication costs in such a way that the total cost of generating global association rules must be less than the cost of combining datasets of all participating sites into a centralized site.

BACKGROUND

DARM aims to discover rules from different datasets that are distributed across multiple sites and intercon-

nected by a communication network. It tries to avoid the communication cost of combining datasets into a centralized site, which requires large amounts of network communication. It offers a new technique to discover knowledge or patterns from such loosely coupled distributed datasets and produces global rule models by using minimal network communication. Figure 1 illustrates a typical DARM framework. It shows three participating sites, where each site generates local models from its respective data repository and exchanges local models with other sites in order to generate global models.

Typically, rules generated by DARM algorithms are considered interesting, if they satisfy both minimum global support and confidence threshold. To find interesting global rules, DARM generally has two distinct tasks: (i) global support count, and (ii) global rules generation.

Let D be a virtual transaction dataset comprised of $D_1, D_2, D_3 \ldots\ldots D_m$ geographically distributed datasets; let n be the number of items and I be the set of items such that $I = \{a_1, a_2, a_3, \ldots\ldots\ldots a_n\}$, where $a_i \subset n$. Suppose N is the total number of transactions and $T = \{t_1, t_2, t_3, \ldots\ldots\ldots t_N\}$ is the sequence of transaction, such that $t_i \subset D$. The support of each element of I is the number of transactions in D containing I and for a given itemset $A \subset I$; we can define its support as follows:

$$Support(A) = \frac{A \subseteq t_i}{N} \quad (1)$$

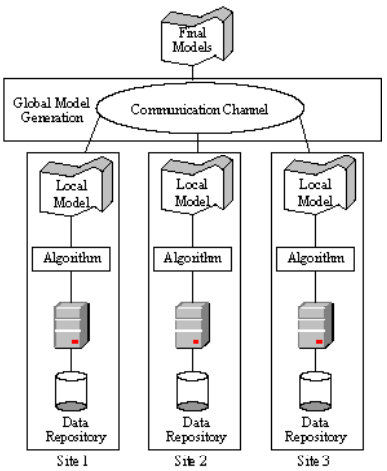
Itemset A is frequent if and only if $Support(A) \geq minsup$, where $minsup$ is a user-defined global support threshold. Once the algorithm discovers all global frequent itemsets, each site generates global rules that have user-specified confidence. It uses frequent itemsets to find the confidence of a rule $R1$ and can be calculated by using the following formula:

$$Confidence(R) = \frac{Support(F_1 \cup F_2)}{Support(F_1)} \quad (2)$$

CHALLENGES OF DARM

All of the DARM algorithms are based on sequential association mining algorithms. Therefore, they inherit all drawbacks of sequential association mining. However, DARM not only deals with the drawbacks of it but also considers other issues related to distributed computing. For example, each site may have different platforms and datasets, and each of those datasets may have different schemas. In the following paragraphs, we discuss a few of them.

Figure 1. A distributed data mining framework



Frequent Itemset Enumeration

Frequent itemset enumeration is one of the main association rule mining tasks (Agrawal & Srikant, 1993; Zaki, 2000; Jiawei, Jian & Yiwen, 2000). Association rules are generated from frequent itemsets. However, enumerating all frequent itemsets is computationally expansive. For example, if a transaction of a database contains 30 items, one can generate up to 2^{30} itemsets. To mitigate the enumeration problem, we found two basic search approaches in the data-mining literature. The first approach uses breadth-first searching techniques that search through the iterate dataset by generating the candidate itemsets. It works efficiently when user-specified support threshold is high. The second approach uses depth-first searching techniques to enumerate frequent itemsets. This search technique performs better when user-specified support threshold is low, or if the dataset is dense (i.e., items frequently occur in transactions). For example, Eclat (Zaki, 2000) determines the support of k-itemsets by intersecting the tidlists (Transaction ID) of the lexicographically first two (k-1) length subsets that share a common prefix. However, this approach may run out of main memory when there are large numbers of transactions.

However, DARM datasets are spread over various sites. Due to this, it cannot take full advantage of those searching techniques. For example, breath-first search performs better when support threshold is high. On the other hand, in DARM, the candidate itemsets are generated by combining frequent items of all datasets; hence, it enumerates those itemsets that are not frequent in a particular site. As a result, DARM cannot utilize the advantage of breath-first techniques when user-specified support threshold is high. In contrast, if a depth-first search technique is employed in DARM, then it needs large amounts of network communication. Therefore, without very fast network connection, depth-first search is not feasible in DARM.

Communication

A fundamental challenge for DARM is to develop mining techniques without communicating data from various sites unnecessarily. Since, in DARM, each site shares its frequent itemsets with other sites to generate unambiguous association rules, each step of DARM requires communication. The cost of communication increases proportionally to the size of candidate itemsets. For example, suppose there are three sites, such as S1, S2, and S3, involved in distributed association mining. Suppose that after the second pass, site S1 has candidate 2-itemsets equal to {AB, AC, BC, BD}, S2 = {AB, AD, BC, BD}, and S3 = {AC, AD, BC, BD}. To generate global frequent 2-itemsets, each site sends its respective candidate 2-itemsets to other sites and receives candidate 2-itemsets from others. If we calculate the total number of candidate 2-itemsets that each site receives, it is equal to 8. But, if each site increases the number of candidate 2-itemsets by 1, each site will receive 10 candidate 2-itemsets, and, subsequently, this increases communication cost. This cost will further increase when the number of sites is increased. Due to this reason, message optimization becomes an integral part of DARM algorithms.

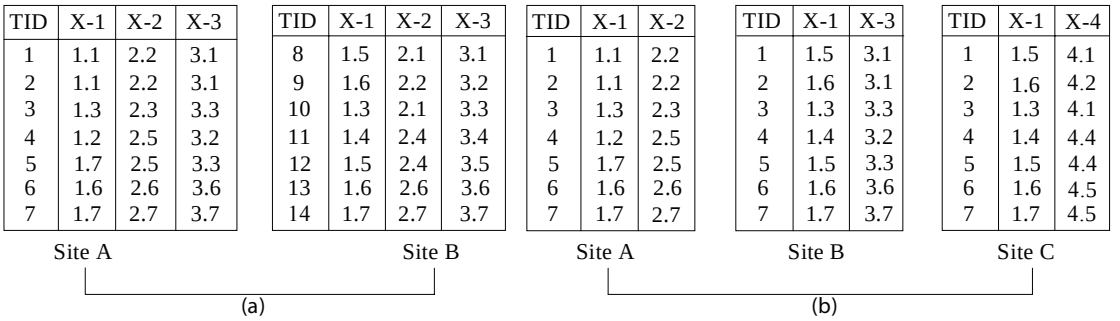
The basic message exchange technique (Agrawal & Shafer, 1996) incurs massive communication costs, especially when the number of local frequent itemsets is large. To reduce message exchange cost, we found several message optimization techniques (Ashrafi et al., 2004; Assaf & Ron, 2002; Cheung et al., 1996). Each of these optimizations focuses on the message exchange size and tries to reduce it in such a way that the overall communication cost is less than the cost of merging all datasets into a single site. However, those optimization techniques are based on some assumptions and, therefore, are not suitable in many situations. For example, DMA (Cheung et al., 1996) assumes the number of disjoint itemsets among different sites is high. But this is only achievable when the different participating sites have vertical fragmented datasets. To mitigate these problems, we found another framework (Ashrafi et al., 2004) in data-mining literature. It partitions the different participating sites into two groups, such as sender and receiver, and uses itemsets history to reduce message exchange size. Each sender site sends its local frequent itemsets to a particular receiver site. When receiver sites receive all local frequent itemsets, it generate global frequent itemsets for that iteration and sends back those itemsets to all sender sites

Privacy

Association rule-mining algorithms discover patterns from datasets based on some statistical measurements. However those statistical measurements have significant meaning and may breach privacy (Evfimievski, Srikant, Agrawal & Gehrke, 2002; Rizvi & Haritsa, 2002). Therefore, privacy becomes a key issue of association rule mining. Distributed association rule-mining algorithms discover association rules beyond the organization boundary. For that reason, the chances of breaching privacy in distributed association mining are higher than the centralized association mining (Vaidya & Clifton, 2002; Ashrafi, David & Kate, 2003), because distributed association mining accomplishes the final rule model by combining various local patterns. For example, there are three sites, S1, S2, and S3, each of which has datasets DS1, DS2, and DS3. Suppose A and B are two items having global support threshold; in order to find rule $A \rightarrow B$ or $B \rightarrow A$, we need to aggregate the local support of itemsets AB from all participating sites (i.e., site S1, S2, and S3). When we do such aggregation, all sites learn the exact support count of other sites. However, participating sites generally are reluctant to disclose the exact support of itemset AB to other sites, because support counts of an itemset has a statistical meaning, and this may thread data privacy.

For the above-mentioned reason, we need secure multi-party computation solutions to maintain privacy of distributed association mining (Kantercioglu & Clifton, 2002). The goal of secure multi-party computation

Figure 2. Distributed (a) homogeneous (b) heterogeneous dataset



TID	X-1	X-2	X-3
1	1.1	2.2	3.1
2	1.1	2.2	3.1
3	1.3	2.3	3.3
4	1.2	2.5	3.2
5	1.7	2.5	3.3
6	1.6	2.6	3.6
7	1.7	2.7	3.7

Site A

TID	X-1	X-2	X-3
8	1.5	2.1	3.1
9	1.6	2.2	3.2
10	1.3	2.1	3.3
11	1.4	2.4	3.4
12	1.5	2.4	3.5
13	1.6	2.6	3.6
14	1.7	2.7	3.7

Site B

(a)

TID	X-1	X-2
1	1.1	2.2
2	1.1	2.2
3	1.3	2.3
4	1.2	2.5
5	1.7	2.5
6	1.6	2.6
7	1.7	2.7

Site A

TID	X-1	X-3
1	1.5	3.1
2	1.6	3.1
3	1.3	3.3
4	1.4	3.2
5	1.5	3.3
6	1.6	3.6
7	1.7	3.7

Site B

TID	X-1	X-4
1	1.5	4.1
2	1.6	4.2
3	1.3	4.1
4	1.4	4.4
5	1.5	4.4
6	1.6	4.5
7	1.7	4.5

Site C

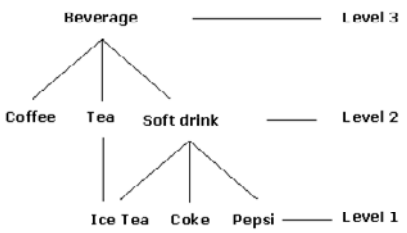
(b)

(SMC) in distributed association rule mining is to find global support of all itemsets using a function where multiple parties hold their local support counts, and, at the end, all parties know about the global support of all itemsets but nothing more. And finally, each participating site uses this global support for rule generation.

Partition

DARM deals with different possibilities of data distribution. Different sites may contain horizontally or vertically partitioned data. In horizontal partition, the dataset of each participating site shares a common set of attributes. However, in vertical partition, the datasets of different sites may have different attributes. Figure 2 illustrates horizontal and vertical partition dataset in a distributed context. Generating rules by using DARM becomes more difficult when different participating sites have vertically partitioned datasets. For example, if a participating site has a subset of items of other sites, then that site may have only a few frequent itemsets and may finish the process earlier than others. However, that site cannot move to the next iteration, because it needs to wait for other sites to generate global frequent itemsets of that iteration.

Figure 3. A generalization scenario



Furthermore, the problem becomes more difficult when the dataset of each site does not have the same hierarchical taxonomy. If a different taxonomy level exists in datasets of different sites, as shown in Figure 3, it becomes very difficult to maintain the accuracy of global models.

FUTURE TRENDS

The DARM algorithms often consider the datasets of various sites as a single virtual table. On the other hand, such assumptions become incorrect when DARM uses different datasets that are not from the same domain. Enumerating rules using DARM algorithms on such datasets may cause a discrepancy, if we assume that semantic meanings of those datasets are the same. The future DARM algorithms will investigate how such datasets can be used to find meaningful rules without increasing the communication cost.

CONCLUSION

The widespread use of computers and the advances in database technology have provided a large volume of data distributed among various sites. The explosive growth of data in databases has generated an urgent need for efficient DARM to discover useful information and knowledge. Therefore, DARM becomes one of the active subareas of data-mining research. It not only promises to generate association rules with minimal communication cost, but it also utilizes the resources distributed among different sites efficiently. However,

the acceptability of DARM depends to a great extent on the issues discussed in this article.

REFERENCES

Agrawal, R., Imielinsky, T., & Sawmi, A.N. (1993). Mining association rules between sets of items in large databases. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Washington D.C.

Agrawal, R., & Shafer, J.C. (1996). Parallel mining of association rules. *IEEE Transactions on Knowledge and Data Engineering, 8*(6), 962-969.

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules in large database. *Proceedings of the International Conference on Very Large Databases*, Santiago de Chile, Chile.

Ashrafi, M.Z., Taniar, D., & Smith, K.A. (2003). Towards privacy preserving distributed association rule mining. *Proceedings of the Distributed Computing, Lecture Notes in Computer Science, IWDC'03*, Calcutta, India.

Ashrafi, M.Z., Taniar, D., & Smith, K.A. (2004). Reducing communication cost in privacy preserving distributed association rule mining. *Proceedings of the Database Systems for Advanced Applications, DASFAA'04*, Jeju Island, Korea.

Ashrafi, M.Z., Taniar, D., & Smith, K.A. (2004). ODAM: An optimized distributed association rule mining algorithm. *IEEE Distributed Systems Online*, IEEE.

Assaf, S., & Ron, W. (2002). Communication-efficient distributed mining of association rules. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, California.

Cheung, D.W., Ng, V.T., Fu, A.W., & Fu, Y. (1996a). Efficient mining of association rules in distributed databases. *IEEE Transactions on Knowledge and Data Engineering, 8*(6), 911-922.

Cheung, D.W., Ng, V.T., Fu, A.W., & Fu, Y. (1996b). A fast distributed algorithm for mining association rules. *Proceedings of the International Conference on Parallel and Distributed Information Systems*, Florida.

Evfimievski, A., Srikant, R., Agrawal, R., & Gehrke, J. (2002). Privacy preserving mining association rules. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Edmonton, Canada.

Jiawei, H., Jian, P., & Yiwen, Y. (2000). Mining frequent patterns without candidate generation. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Dallas, Texas.

Kantercioglu, M., & Clifton, C. (2002). Privacy preserving distributed mining of association rules on horizontal partitioned data. *Proceedings of the ACM SIGMOD Workshop of Research Issues in Data Mining and Knowledge Discovery DMKD*, Edmonton, Canada.

Rizvi, S.J., & Haritsa, J.R. (2002). Maintaining data privacy in association rule mining. *Proceedings of the International Conference on Very Large Databases*, Hong Kong, China.

Vaidya, J., & Clifton, C. (2002). Privacy preserving association rule mining in vertically partitioned data. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Edmonton, Canada.

Zaki, M.J. (1999). Parallel and distributed association mining: A survey. *IEEE Concurrency, 7*(4), 14-25.

Zaki, M.J. (2000). Scalable algorithms for association mining. *IEEE Transactions on Knowledge and Data Engineering, 12*(2), 372-390.

Zaki, M.J., & Ya, P. (2002). Introduction: Recent developments in parallel and distributed data mining. *Journal of Distributed and Parallel Databases, 11*(2), 123-127.

KEY TERMS

DARM: Distributed Association Rule Mining.

Data Center: A centralized repository for the storage and management of information, organized for a particular area or body of knowledge.

Frequent Itemset: A set of itemsets that have the user specified support threshold.

Network Intrusion Detection: A system that detects inappropriate, incorrect, or anomalous activity in the private network.

SMC: SMC computes a function $f(x_1, x_2, x_3 \ldots x_n)$ that holds inputs from several parties, and, at the end, all parties know about the result of the function $f(x_1, x_2, x_3 \ldots x_n)$ and nothing else.

Taxonomy: A classification based on a pre-determined system that is used to provide a conceptual framework for discussion, analysis, or information retrieval.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 403-407, copyright 2005 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

Distributed Data Aggregation Technology for Real-Time DDoS Attacks Detection

Yu Chen
State University of New York – Binghamton, USA

Wei-Shinn Ku
Auburn University, USA

INTRODUCTION

The information technology has revolutionized almost every facet of our lives. Government, commercial, and educational organizations depend on computers and Internet to such an extent that day-to-day operations are significantly hindered when the networks are "down" (Gordon, Loeb, Lucyshyn & Richardson, 2005). The prosperity of the Internet also attracted abusers and attackers motivated for personal, financial, or even political reasons. What attackers aim at currently is beyond obtaining unauthorized network accesses or stealing private information, there have been attacks on Internet infrastructures (Chakrabarti & Manimaran, 2002; Moore, Voelker & Savage, 2001; Naoumov & Ross, 2006).

Distributed Denial of Services (DDoS) attacks is one of such attacks that can lead to enormous destruction, as different infrastructure components of the Internet have implicit trust relationship with each other (Mirkovic & Reiher, 2004; Specht & Lee, 2004). The DDoS attacker often exploits the huge resource asymmetry between the Internet and the victim systems (Chen, Hwang & Ku, 2007; Douligeris & Mitrokosta, 2003).

A comprehensive solution to DDoS attacks requires covering global effects over a wide area of *autonomous system* (AS) domains on the Internet (Mirkovic & Reiher, 2005). Timely detection of the ongoing attacks is the prerequisite of any effective defense scheme (Carl, Kesidis, Brooks & Rai, 2006). It is highly desirable to detect DDoS attacks at very early stage, instead of waiting for the flood to become widespread. It is mandatory for the detection systems to collect real time traffic data from widely deployed traffic monitors and construct the spatiotemporal pattern of anomaly propagation inside the network.

This chapter will introduce a novel distributed real time data aggregation technique named *Change Aggregation Tree* (CAT). The CAT system adopts a hierarchical architecture to simplify the alert correlation and global detection procedures. At intra-domain level, each individual router, which plays the role of traffic monitor, periodically report the local traffic status to the CAT server in the AS. At the inter-domain layer, CAT servers share local detected anomaly patterns with peers located in other ASes, where the potential attack victim is located.

BACKGROUND

To monitor the traffic fluctuations in a real time manner, network devices often play the role of distributed sensor system that collects local data individually. However, as a large scale distributed system without a central administrator, it is challenging to create a spatiotemporal picture covering wide area cross multiple ISP networks. Unfortunately, such a big picture is essential to detect the anomalies embedded in the traffic flows (Chen, Hwang & Ku, 2007; Papadopoulos, Lindell, Mehringer, Hussain & Govindan, 2003). For this reason, efficient distributed data aggregation techniques have become a hot topic in research community. Due to the limited space, here we only provide a brief survey of reported works which are closely relevant to our work.

A couple of overlay based data aggregation techniques have been proposed to monitor local network traffic and detect anomalies and attacks collaboratively (Feinstein, Schnackenberg, Balupari & Kindred, 2003). In WormShield (Cai, Hwang, Pan & Papadopoulos, 2007), a balanced distributed data aggregation tree (DAT) was proposed, which is capable of collecting and aggregating the fingerprint of Internet worms generated locally. Comparing to the original overlay based data aggregation such as Chord (Stoica, Morris, Karger, Kaashoek & Balakrishnan, 2001), DAT

can compute global fingerprint statistics in a scalable and load-balanced fashion. Several data aggregation systems use advanced statistical algorithms to predict lost values (Zhao, Govindan & Estrin, 2003; Madden, Franklin, Hellerstein & Hong, 2002) and try to reduce the sensitivity of large scale data aggregation networks to the loss of data (Huang, Zhao, Joseph & Kubiatowicz, 2006).

MAIN FOCUS

Efficient distributed data aggregation technique is critical to monitor the spatiotemporal fluctuations in Internet traffic status. Chen, Hwang and Ku (2007) have proposed a new distributed aggregation scheme based on change-point detection across multiple network domains. In order to establish an early warning system for DDoS defense across multiple domains, this system adopts a new mechanism called *change aggregation tree* (CAT), which adopts a hierarchical architecture and simplifies the alert correlation and global detection procedures implemented in ISP core networks.

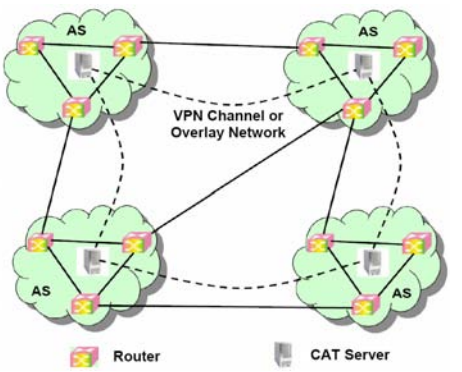
Distributed Change Point Detection

The *Distributed Change-point Detection* (DCD) scheme detects DDoS flooding attacks by monitoring the propagation patterns of abrupt traffic changes at distributed network points. Once a sufficiently large CAT tree is constructed to exceed a preset threshold, an attack is declared. Figure 1 presents the system architecture of the DCD scheme. The system is deployed over multiple AS domains. There is a central CAT server in each domain. The system detects traffic changes, checks flow propagation patterns, aggregates suspicious alerts, and merge CAT subtrees from collaborative servers into a global CAT tree. The root of the global CAT tree is at the victim end. Each tree node corresponds to an *attack-transit routers* (ATR). Each tree edge corresponds to a link between the attack-transit routers.

The DCD system has hierarchical detection architecture. There are three layers in this architecture as shown in Fig. 2. At the lowest layer, individual router functions as a sensor to monitor local traffic fluctuations. Considering the directionality and homing effects in a DDoS flooding attack, routers check how the wavefront changes. A router raises an alert and reports an anomalous traffic pattern to the CAT server. The second layer is at each network domain level. The CAT server constructs a CAT subtree that displays a spatiotemporal pattern of the attack flow in the domain. At the highest layer, the CAT servers at different domains form an overlay network or communicate with each other through *virtual private network* (VPN) channels. All CAT servers send the locally-generated CAT subtrees to

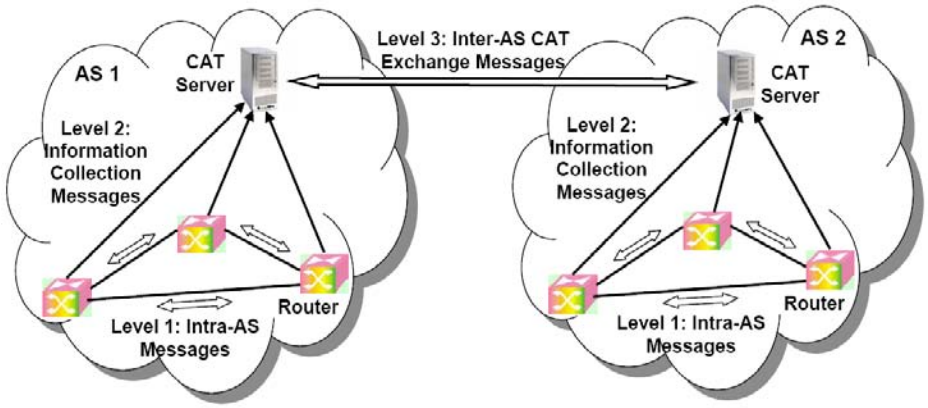
Figure 1. Distributed change detection of DDoS attacks over multiple AS domains

(a) Multi-domain DDoS defense system



(b) Inter-domain communication via VPN tunnels or overlay network atop the CAT servers in 4 domains

Figure 2. Illustration of 3-level communication between two CAT servers in two AS domains



the server in the destination domain, where the victim is attached. By merging CAT subtrees from cooperative domains, the destination server has a global picture of the attack. The larger is the global CAT tree constructed, the higher is the threat experienced.

Global CAT Aggregation

This section presents the procedure of global change data aggregation. Starting from the single domain subtree construction, we will discuss the global CAT tree merging and analyze the complexity based on real life Internet domain distribution data set.

Constructing Subtrees at Domain Servers

The router reports the identifier of a flow causing the traffic surge. Since all routers are under the same ISP authority and work cooperatively, each router knows their immediate neighbors. The reported message provides the upstream and downstream router identifiers. The main purpose of sending the flow status message is to report where the suspicious flows are captured.

To indicate the location of a suspicious flow, the router identifier must send. We need to identify the flow identifier of the *n*-bit prefix of the destination IP addresses. To construct the CAT, the status report provides the upstream and downstream router identifiers instead of router I/O port numbers. Using the reported status information, the domain server constructs the CAT tree gradually after receiving the alert reports from the ATRs. Table 1 summarizes the information carried in a typical alert message from an ATR.

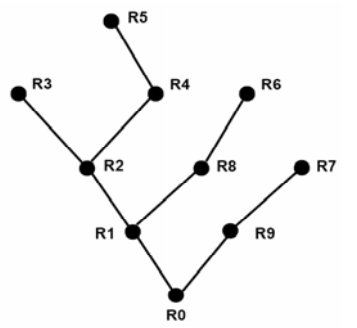
The output of this procedure is a single-domain CAT subtree. Figure 3(a) shows a flooding attack launched from 4 zombies. The ATRs detect abnormal surge of traffic at their I/O ports. The victim is attached with the end router R0 in the Fig. 3(a). All the attack flows form the flow homing towards the end router. Using the data entries listed in Table 1, The CAT tree can be

Table 1. Alert message reported by a router

Parameter	Brief Definition
nd_id	The router ID,
fl_id	The flow ID
up_num	Number of upstream nodes
dn_num	Number of downstream nodes
up_id	node ID of upstream node
dn_id	node ID of downstream node
Router status	Suspicious attack or normal traffic

Figure 3. An example of a single domain subtree construction

(a) An example of traffic flow of a DDoS flooding attack launched from 4 zombies.



(b) A change aggregation tree for the flooding pattern in (a).

specified by a hierarchical data structure. The root node carries the flow ID, the number of routers involved, root node ID, and the count of child nodes at the next level. Figure 3 illustrates how a CAT subtree rooted at the end router is constructed by merging the alert reports from 9 ATRs. The upstream and downstream ATRs report to the CAT server during each monitory cycle.

Global CAT Aggregation

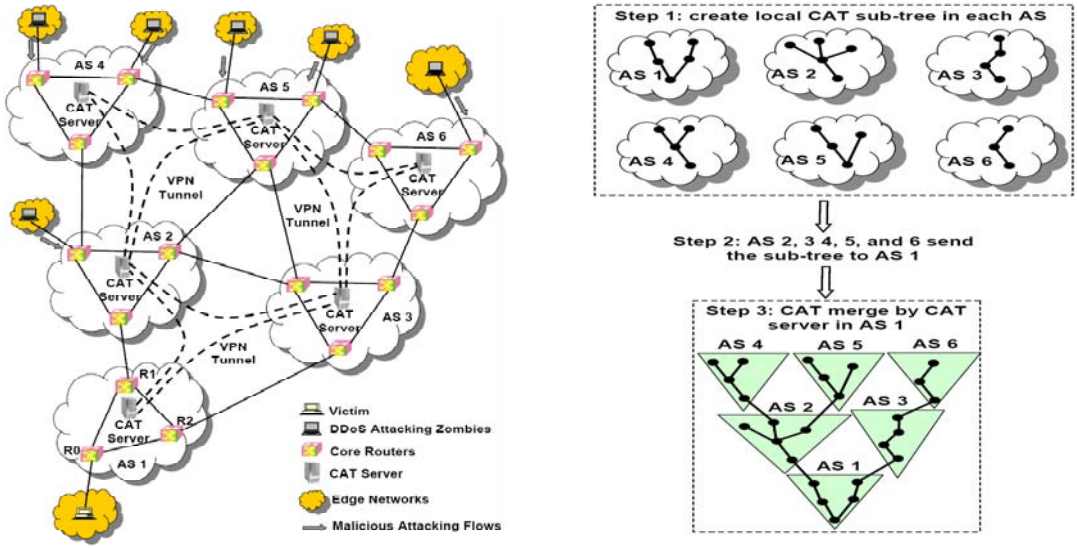
In a DDoS flooding attack, the attacker often recruits many zombies distributed over the Internet. The flooding traffic travels through multiple AS domains before reaching the edge network, where the victim is physically attached. Routers at the upstream domains observe the suspicious traffic flows earlier than routers at the downstream networks. The CAT subtrees constructed at all traversed domains must be merged to yield a global CAT tree at the destination domain. The tree width and height thus reveal the scope of the DDoS attack. On receiving subtrees from upstream CAT servers, the CAT server in the destination domain builds the global CAT tree from its local subtree. Figure 4 shows an example network environment involving six AS domains.

The victim system is located in the AS1 domain. Zombies are scattered widely in Internet outside the illustrated domains. By detecting abnormal traffic changes in each domain, the CAT server creates a CAT subtree locally at each domain. Figure 4(b) shows three steps taken to merge the 6 subtrees generated by 6 CAT servers of 6 AS domain. All 6 subtrees are resulted from checking the packets belonging to the same flow traffic destined for the same domain AS1. Five subtrees generated at AS 2, AS3, AS4, AS5, and AS6 at upstream domains are sent to AS1 at Step 2. Then, the concatenated CAT subtrees are connected to the downstream subtree at AS1. Thus the global CAT tree is finally rooted at the last hop router to an edge network R0 that is attached to the victim system.

Complexity Analysis

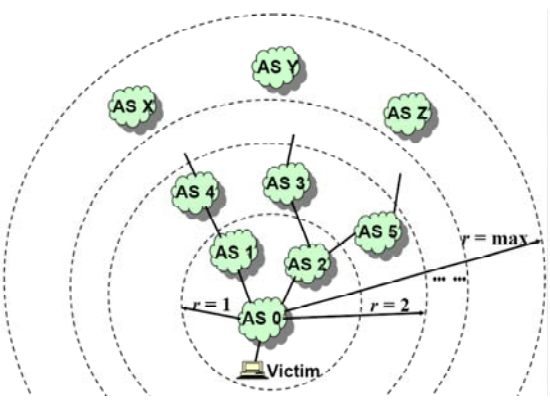
The complexity of the CAT growth is analyzed below based on Internet topology data available from open literature (ISO, 2006; Siganos, M. Faloutsos, P. Faloutsos & C. Faloutsos, 2003). Figure 5 illustrates the process of the CAT tree growth out of merging subtrees from attack-transit domains. Let ***r*** be the number of hops from an AS domain to the destination domain.

Figure 4. An example 6-domain global CAT tree construction environment.



(a) DCD system architecture over 6 domains. *(b) Merging 6 subtrees to yield a global tree*

Figure 5. Merging CAT subtrees from neighboring AS domains to outer domains to build a global CAT tree, where AS_0 is the victim domain and r_{max} = 6 hops



The server checks the received subtrees in increasing order of distance ***r***.

The system first merges the subtrees from ASes located in 1-hop (r = 1) distance to form a partial global tree. Next, it merges the subtrees from domains at 2-hop distance. The merging process repeats with distances r = 3, 4 until all subtrees are merged into the final global CAT tree.

The complexity of global CAT tree merging is highly dependent on the network topology. We model the Internet domains as an undirected graph of M nodes and E edges. The diameter of the graph is denoted by δ. Siganos *et al.* (Siganos, M. Faloutsos, P. Faloutsos & C. Faloutsos, 2003) models the Internet neighborhood as an H-dimensional sphere with a diameter $\boldsymbol{\delta}$. The parameter H is the dimension of the network topology (M. Faloutsos, C. Faloutsos, and P. Faloutsos, 1999). For example, H = 1 specifies a ring topology and H = 2 for a 2-dimensional mesh. Any two nodes are within an effective diameter, δ_{ef} hops away from each other.

In 2002, the dimension of Internet was calculated as H = 5.7 in an average sense. The ceiling of this diameter δ_{ef} is thus set to be 6. Let NN(h) be the number of domains located at distance h from a typical domain in the Internet. Table 2 gives the domain distribution – the probability of an AS domain residing exactly h hops away from a reference domain. The numbers of domains in various distance ranges are given in the

Table 2. Internet domain distribution reported in Feb. 28, 2006 (ISO, 2006)

Hop Count, h	1	2	3	4	5	6	≥ 7
Domain Distribution, p_h	0.04%	8.05%	38.55%	38.12%	12.7%	2.25%	0.29%
Domain Count, $NN(h)$	14	2,818	13,493	13,342	4,445	788	102

second row. It is interesting to note that most communicating domains are within 3 or 4 hops, almost following a normal distribution centered on an average hop count of 3.5.

The number of Internet AS domains keeps increasing in time, the Faloutsos reports (M. Faloutsos, C. Faloutsos, and P. Faloutsos, 1999; Siganos, M. Faloutsos, P. Faloutsos & C. Faloutsos, 2003) indicates that this AS distribution is pretty stable over time. This implies that a packet can reach almost all domains in the Internet by traversing through 6 hops. Therefore, we set the maximum hop count $r_{max} = 6$ in Figure 5.

FUTURE TRENDS

One of the most critical concerns of the real time distributed data aggregation for the security purpose of the Internet traffic monitoring is the scalability. There is limited resources in core routers can be shared to execute security functions. Considering the computational complexity, it is challenging to cope with the high data rate in today's network (multi gigabyte). Especially, it is desired to detect the attacks swiftly before damages caused. In general, most software based security mechanisms are not feasible in high-speed core networks.

We suggest exploring a hardware approach to implementing the CAT mechanism and other real time distributed data aggregation techniques using reconfigurable hardware devices, for instance, network processors or FPGA devices. The ultimate goal is to promote real-time detection and response against DDoS attacks with automatic signature generation. FPGA devices have the flexibility of software combined with a speed of a hardware implementation, providing high performance and fast development cycles. The reconfigurability allows further update of the security system considering the evolution of attacks and defense techniques. The high parallelism makes it capable of handling multi-gigabyte of data rate in the core networks. It pushes security tasks down to the lower level in the packet-processing hierarchy. The suspicious attacking traffic patterns are recognized before the traffic hit the routing fabric. The security monitoring works are performed in parallel with routing jobs and light overhead is incurred. For this purpose, researches on embedded accelerators using network processors or FPGAs have been carried out and solid achievements have been made in recent years (Attig & Lockwood, 2005; Charitakis, Anagnostakis & Markatos, 2004).

CONCLUSION

It is crucial to detect the DDoS flooding attacks at their early launching stage before widespread damages done to legitimate applications on the victim system. We develop a novel distributed data aggregation scheme to monitor the spatiotemporal distribution of changes in Internet traffic. This CAT mechanism helps us to achieve an early DDoS detection architecture based on constructing a global change aggregation tree that describes the propagation of the wavefront of the flooding attacks.

REFERENCES

Attig, M., & Lockwood, J. (2005). A Framework For Rule Processing in Reconfigurable Network Systems. *Proc. of IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM)*, Napa, CA., April 17-20, 2005;

Cai, M., Hwang, K., Pan, J., & Papadupolous, C. (2007). WormShield: Fast Worm Signature Generation with Distributed Fingerprint Aggregation. *IEEE Trans. of Dependable and Secure Computing*, Vol.4, No. 2, April/June 2007.

Carl, G., Kesidis, G., Brooks, R., & Rai, S. (2006). Denial-of-Service Attack Detection Techniques. *IEEE Internet Computing*, January/February 2006.

Chakrabarti, A., & Manimaran, G. (2002). Internet Infrastructure Security: A Taxonomy. *IEEE Network*, November 2002.

Charitakis, I., Anagnostakis, K., & Markatos, E. P. (2004). A Network-Professor-Based Traffic Splitter for Intrusion Detection. *ICS-FORTH Technical Report 342*, September 2004.

Chen, Y., Hwang, K., & Ku, W. (2007). Collaborative Monitoring and Detection of DDoS Flooding Attacks in ISP Core Networks. *IEEE Transaction on Parallel and Distributed Systems*, Vol. 18, No. 12, December 2007.

Douligeris, C., & Mitrokosta, A. (2003). DDoS Attacks and Defense Mechanisms: a Classification. Proceedings of the 3rd IEEE International Symposium on Signal Processing and Information Technology, 2003, (ISSPIT 2003).

Faloutsos, M., Faloutsos, C., & Faloutsos, P. (1999). On Power-law Relationships of the Internet Topology. *Proc. of ACM SIGCOMM*, Aug. 1999.

Feinstein, L., Schnackenberg, D., Balupari, R., & Kindred, D. (2003). Statistical Approaches to DDoS Attack Detection and Response. Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX03), Washington, DC.

Gibson, S. (2002). Distributed Reflection Denial of Service. Feb. 22, 2002, http://grc.com/dos/drdos.htm;

Gordon, L., Loeb, M., Lucyshyn, W., & Richardson, R. (2005). 10th Annual CSI/FBI Computer Crime and Security Survey. *Computer Security Institute* (CSI).

Huang, L., Zhao, B. Y., Joseph, A. D., & Kubiatowicz, J. (2006). Probabilistic Data Aggregation In Distributed Networks. *Technical Report No. UCB/EECS-2006-11*, Electrical Engineering and Computer Sciences, University of California at Berkeley, Feb. 6, 2006.

ISO 3166 Report (2006). AS Resource Allocations. http://bgp.potaroo.net/iso3166/ascc.html

Madden, S., Franklin, M. J., Hellerstein, J. M., & Hong, W. (2002). TAG: a Tiny AGgregation service for ad-hoc sensor networks. *Proc. of OSDI*, 2002.

Mirkovic J., & Reiher, P. (2004). A Taxonomy of DDoS Attack and DDoS Defence Mechanisms. *ACM Computer Communications Review*, vol. 34, no. 2, April 2004.

Mirkovic J., & Reiher, P. (2005). D-WARD: A Source-End Defense Against Flooding DoS Attacks. *IEEE Trans. on Dependable and Secure Computing*, July 2005.

Moore, D., Voelker, G., & Savage, S. (2001). Inferring Internet Denial-of-Service Activity. *Proc. of the 10th USENIX Security Symposium*.

Naoumov, N., & Ross, K. (2006). Exploiting P2P Systems for DDoS Attacks. *International Workshop on Peer-to-Peer Information Management (keynote address)*, Hong Kong, May 2006.

Papadopoulos, C., Lindell, R., Mehringer, J., Hussain, A., & Govindan, R. (2003). COSSACK: Coordinated Suppression of Simultaneous Attacks. *Proc. of DISCEX III*, 2003.

Siganos, G., Faloutsos, M., Faloutsos, P., & Faloutsos, C. (2003). Power-Laws and the AS-level Internet Topology. *ACM/IEEE Trans. on Networking*, pp. 514-524.

Specht, S. M., & Lee, R. B. (2004). Distributed Denial of Service: Taxonomies of Attacks, Tools and Countermeasures. *Proc. of Parallel and Dist. Comp. Systems*, San Francisco, Sept. 15-17, 2004.

Stoica, I., Morris, R., Karger, D., Kaashoek, F., & Balakrishnan, H. (2001). Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications. *Proc. 2001 ACM Conf. Applications, Technologies, Architectures, and Protocols for Computer Comm.* (SIGCOMM), 2001.

Zhao, J., Govindan, R., & Estrin, D. (2003). Computing aggregates for monitoring wireless sensor networks. *Proc. of SNPA*, 2003.

KEY TERMS

Attack Transit Router (ATR): Routers located in the path through which malicious attacking traffic go through towards the victim.

Autonomous Systems: An AS is a connected group of one or more Internet Protocol prefixes run by one or more network operators which has a SINGLE and CLEARLY DEFINED routing policy.

Change Aggregation Tree (CAT): A distributed data structure that describes the spatiotemporal pattern of anomaly traffic surges in a wide area in the network.

Data Aggregation: Data aggregation is any process in which information is gathered and expressed in a summary form, for purposes such as statistical analysis.

DDoS Attacks: On the Internet, a distributed denial-of-service (DDoS) attack is one in which a multitude of compromised systems attack a single target, thereby causing denial of service for users of the targeted system. The flood of incoming messages to the target system essentially forces it to shut down, thereby denying service to the system to legitimate users.

Distributed Change-Point Detection (DCD): A distributed statistical scheme that is design to detect anomalies in a network by monitoring the short term change in the network traffic and comparing it with the long term history average.

FPGA: A field-programmable gate array (FPGA) is an integrated circuit (IC) that can be programmed in the field after manufacture.

Internet: The global internetwork based on the Internet (TCP/IP) architecture, connecting millions of hosts worldwide.

Network Domain: On the Internet, a domain consists of a set of network addresses. This domain is organized in levels. The top level identifies geographic or purpose commonality. The second level identifies a unique place within the top level domain and is, in fact, equivalent to a unique address on the Internet (an IP address). Lower levels of domain may also be used.

Network Processor: A device that is similar to a microprocessor, except that it has been optimized for use in applications involving network routing and packet processing. There is no standard architecture, but many network processors feature multiple RISC CPUs running in parallel. In this configuration, one central processor typically receives and handles network control packets while the others pass data packets through the system at network speeds.

Virtual Private Network (VPN): A communications network tunneled through another network, and dedicated for a specific network. One common application is secure communications through the public Internet, but a VPN need not have explicit security features, such as authentication or content encryption.

Zombie: Zombies are computers that have been compromised by attackers generally through the use of Trojans. Collectively, they are manipulated to create the high traffic flow necessary to create a DDoS attack. In literatures, they are also referred as *agents*.

Distributed Data Mining

Grigorios Tsoumakas
Aristotle University of Thessaloniki, Greece

Ioannis Vlahavas
Aristotle University of Thessaloniki, Greece

INTRODUCTION

The continuous developments in information and communication technology have recently led to the appearance of distributed computing environments, which comprise several, and different sources of large volumes of data and several computing units. The most prominent example of a distributed environment is the Internet, where increasingly more databases and data streams appear that deal with several areas, such as meteorology, oceanography, economy and others. In addition the Internet constitutes the communication medium for geographically distributed information systems, as for example the earth observing system of NASA (*eos.gsfc.nasa.gov*). Other examples of distributed environments that have been developed in the last few years are *sensor networks* for process monitoring and *grids* where a large number of computing and storage units are interconnected over a high-speed network.

The application of the classical knowledge discovery process in distributed environments requires the collection of distributed data in a data warehouse for central processing. However, this is usually either ineffective or infeasible for the following reasons:

(1) *Storage cost*. It is obvious that the requirements of a central storage system are enormous. A classical example concerns data from the astronomy science, and especially images from earth and space telescopes. The size of such databases is reaching the scale of exabytes (10^{18} bytes) and is increasing at a high pace. The central storage of the data of all telescopes of the planet would require a huge data warehouse of enormous cost.
(2) *Communication cost*. The transfer of huge data volumes over network might take extremely much time and also require an unbearable financial cost. Even a small volume of data might create problems in wireless network environments with limited bandwidth. Note also that communication may be a continuous overhead, as distributed databases are not always constant and unchangeable. On the contrary, it is common to have databases that are frequently updated with new data or data streams that constantly record information (e.g remote sensing, sports statistics, etc.).
(3) *Computational cost*. The computational cost of mining a central data warehouse is much bigger than the sum of the cost of analyzing smaller parts of the data that could also be done in parallel. In a grid, for example, it is easier to gather the data at a central location. However, a distributed mining approach would make a better exploitation of the available resources.
(4) *Private and sensitive data*. There are many popular data mining applications that deal with sensitive data, such as people's medical and financial records. The central collection of such data is not desirable as it puts their privacy into risk. In certain cases (e.g. banking, telecommunication) the data might belong to different, perhaps competing, organizations that want to exchange knowledge without the exchange of raw private data.

This article is concerned with Distributed Data Mining algorithms, methods and systems that deal with the above issues in order to discover knowledge from distributed data in an effective and efficient way.

BACKGROUND

Distributed Data Mining (DDM) (Fu, 2001; Park & Kargupta, 2003) is concerned with the application of the classical Data Mining procedure in a distributed computing environment trying to make the best of the available resources (communication network, computing units and databases). Data Mining takes place both

locally at each distributed site and at a global level where the local knowledge is fused in order to discover global knowledge.

A typical architecture of a DDM approach is depicted in Figure 1. The first phase normally involves the analysis of the local database at each distributed site. Then, the discovered knowledge is usually transmitted to a merger site, where the integration of the distributed local models is performed. The results are transmitted back to the distributed databases, so that all sites become updated with the global knowledge. In some approaches, instead of a merger site, the local models are broadcasted to all other sites, so that each site can in parallel compute the global model.

Distributed databases may have homogeneous or heterogeneous schemata. In the former case, the attributes describing the data are the same in each distributed database. This is often the case when the databases belong to the same organization (e.g. local stores of a chain). In the latter case the attributes differ among the distributed databases. In certain applications a key attribute might be present in the heterogeneous databases, which will allow the association between tuples. In other applications the target attribute for prediction might be common across all distributed databases.

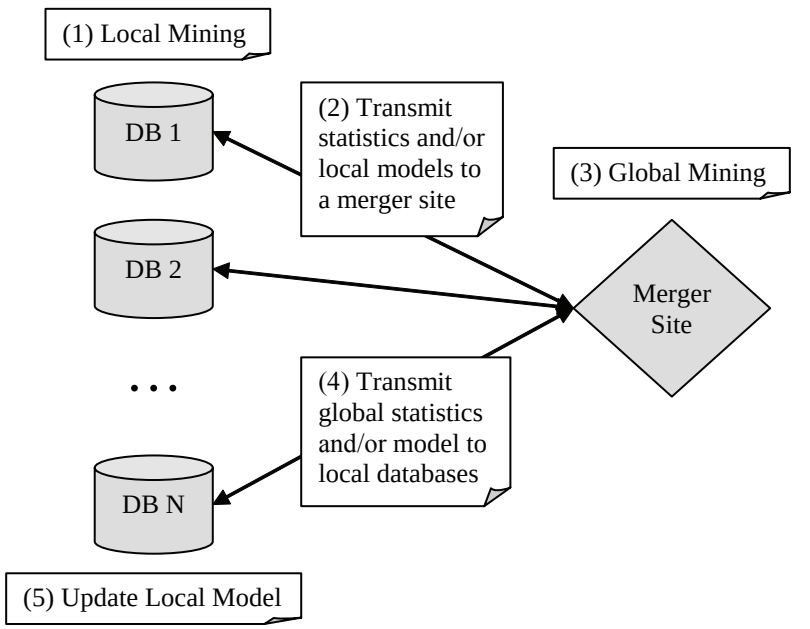
MAIN FOCUS

Distributed Classification and Regression

Approaches for distributed classification and regression are mainly inspired from methods that appear in the area of ensemble methods, such as Stacking, Boosting, Voting and others. Some distributed approaches are straightforward adaptations of ensemble methods in a distributed computing environment, while others extend the existing approaches in order to minimize the communication and coordination costs that arise.

Chan and Stolfo (1993) applied the idea of Stacked Generalization (Wolpert, 1992) to DDM via their meta-learning methodology. They focused on combining distributed data sets and investigated various schemes for structuring the meta-level training examples. They showed that meta-learning exhibits better performance with respect to majority voting for a number of domains. Knowledge Probing (Guo & Sutiwaraphun, 1999) builds on the idea of meta-learning and in addition uses an independent data set, called the probing set, in order to discover a comprehensible model. The output of a meta-learning system on this independent data set together with the attribute value vector of the same

Figure 1. Typical architecture of Distributed Data Mining approaches



data set are used as training examples for a learning algorithm that outputs a final model.

The Collective Data Mining (CDM) framework (Kargupta, Park, Hershberger & Johnson, 2000) allows the learning of classification and regression models over heterogeneous databases. It is based on the observation that any function can be represented using an appropriate set of basis functions. Initially, CDM generates approximate orthonormal basis coefficients at each site. It then moves an appropriately chosen sample of each data set to a single site and generates the approximate basis coefficients corresponding to non-linear cross terms. Finally, it combines the local models and transforms the global model into the user-specified canonical representation.

A number of approaches have been presented for learning a single rule set from distributed data. Hall, Chawla and Bowyer (1997; 1998) present an approach that involves learning decision trees in parallel from disjoint data, converting trees to rules and then combining the rules into a single rule set. Hall, Chawla, Bowyer and Kegelmeyer (2000) present a similar approach for the same case, with the difference that rule learning algorithms are used locally. In both approaches, the rule combination step starts by taking the union of the distributed rule sets and continues by resolving any conflicts that arise. Cho and Wüthrich (2002) present a different approach that starts by learning a single rule for each class from each distributed site. Subsequently, the rules of each class are sorted according to a criterion that is a combination of confidence, support and deviation, and finally the top k rules are selected to form the final rule set. Conflicts that appear during the classification of new instances are resolved using the technique of relative deviation (Wüthrich, 1997).

Fan, Stolfo and Zhang (1999) present d-sampling AdaBoost, an extension to the generalized AdaBoost learning algorithm (Schapire and Singer, 1999) for DDM. At each round of the algorithm, a different site takes the role of training a weak model using the locally available examples weighted properly so as to become a distribution. Then, the update coefficient α_t is computed based on the examples of all distributed sites and the weights of all examples are updated. Experimental results show that the performance of the proposed algorithm is in most cases comparable to or better than learning a single classifier from the union of the distributed data sets, but only in certain cases comparable to boosting that single classifier. The distributed boosting algorithm of Lazarevic and Obradovic (2001) at each round learns a weak model in each distributed site in parallel. These models are exchanged among the sites in order to form an ensemble, which takes the role of the hypothesis. Then, the local weight vectors are updated at each site and their sums are broadcasted to all distributed sites. This way each distributed site maintains a local version of the global distribution without the need of exchanging the complete weight vector. Experimental results show that the proposed algorithm achieved classification accuracy comparable or even slightly better than boosting on the union of the distributed data sets.

Distributed Association Rule Mining

Agrawal and Shafer (1996) discuss three parallel algorithms for mining association rules. One of those, the Count Distribution (CD) algorithm, focuses on minimizing the communication cost, and is therefore suitable for mining association rules in a distributed computing environment. CD uses the Apriori algorithm (Agrawal and Srikant, 1994) locally at each data site. In each pass k of the algorithm, each site generates the same candidate k-itemsets based on the globally frequent itemsets of the previous phase. Then, each site calculates the local support counts of the candidate itemsets and broadcasts them to the rest of the sites, so that global support counts can be computed at each site. Subsequently, each site computes the k-frequent itemsets based on the global counts of the candidate itemsets. The communication complexity of CD in pass k is $O(|C_k|n^2)$, where C_k is the set of candidate k-itemsets and n is the number of sites. In addition, CD involves a synchronization step when each site waits to receive the local support counts from every other site.

Another algorithm that is based on Apriori is the Distributed Mining of Association rules (DMA) algorithm (Cheung, Ng, Fu & Fu, 1996), which is also found as Fast Distributed Mining of association rules (FDM) algorithm in (Cheung, Han, Ng, Fu & Fu, 1996). DMA generates a smaller number of candidate itemsets than CD, by pruning at each site the itemsets that are not locally frequent. In addition, it uses polling sites to optimize the exchange of support counts among sites, reducing the communication complexity in pass k to $O(|C_k|n)$, where C_k is the set of candidate k-itemsets and n is the number of sites. However, the performance enhancements of DMA over CD are based on the as-

sumption that the data distributions at the different sites are skewed. When this assumption is violated, DMA actually introduces a larger overhead than CD due to its higher complexity.

The Optimized Distributed Association rule Mining (ODAM) algorithm (Ashrafi, Taniar & Smith, 2004) follows the paradigm of CD and DMA, but attempts to minimize communication and synchronization costs in two ways. At the local mining level, it proposes a technical extension to the Apriori algorithm. It reduces the size of transactions by: i) deleting the items that weren't found frequent in the previous step and ii) deleting duplicate transactions, but keeping track of them through a counter. It then attempts to fit the remaining transaction into main memory in order to avoid disk access costs. At the communication level, it minimizes the total message exchange by sending support counts of candidate itemsets to a single site, called receiver. The receiver broadcasts the globally frequent itemsets back to the distributed sites.

Distributed Clustering

Johnson and Kargupta (1999) present the Collective Hierarchical Clustering (CHC) algorithm for clustering distributed heterogeneous data sets, which share a common key attribute. CHC comprises three stages: i) local hierarchical clustering at each site, ii) transmission of the local dendrograms to a facilitator site, and iii) generation of a global dendrogram. CHC estimates a lower and an upper bound for the distance between any two given data points, based on the information of the local dendrograms. It then clusters the data points using a function on these bounds (e.g. average) as a distance metric. The resulting global dendrogram is an approximation of the dendrogram that would be produced if all data were gathered at a single site.

Samatova, Ostrouchov, Geist and Melechko (2002), present the RACHET algorithm for clustering distributed homogeneous data sets. RACHET applies a hierarchical clustering algorithm locally at each site. For each cluster in the hierarchy it maintains a set of descriptive statistics, which form a condensed summary of the data points in the cluster. The local dendrograms along with the descriptive statistics are transmitted to a merging site, which agglomerates them in order to construct the final global dendrogram. Experimental results show that RACHET achieves good quality of clustering compared to a centralized hierarchical clustering algorithm, with minimal communication cost.

Januzaj, Kriegel and Pfeifle (2004) present the Density Based Distributed Clustering (DBDC) algorithm. Initially, DBDC uses the DBSCAN clustering algorithm locally at each distributed site. Subsequently, a small number of representative points that accurately describe each local cluster are selected. Finally, DBDC applies the DBSCAN algorithm on the representative points in order to produce the global clustering model.

Database Clustering

Real-world, physically distributed databases have an intrinsic data skewness property. The data distributions at different sites are not identical. For example, data related to a disease from hospitals around the world might have varying distributions due to different nutrition habits, climate and quality of life. The same is true for buying patterns identified in supermarkets at different regions of a country. Web document classifiers trained from directories of different Web portals is another example.

Neglecting the above phenomenon, may introduce problems in the resulting knowledge. If all databases are considered as a single logical entity then the idiosyncrasies of different sites will not be detected. On the other hand if each database is mined separately, then knowledge that concerns more than one database might be lost. The solution that several researchers have followed is to cluster the databases themselves, identify groups of similar databases, and apply DDM methods on each group of databases.

Parthasarathy and Ogihara (2000) present an approach on clustering distributed databases, based on association rules. The clustering method used, is an extension of hierarchical agglomerative clustering that uses a measure of similarity of the association rules at each database. McClean, Scotney, Greer and Páircéir (2001) consider the clustering of heterogeneous databases that hold aggregate count data. They experimented with the Euclidean metric and the Kullback-Leibler information divergence for measuring the distance of aggregate data. Tsoumakas, Angelis and Vlahavas (2003) consider the clustering of databases in distributed classification tasks. They cluster the classification models that are produced at each site based on the differences of their predictions in a validation data set. Experimental results show that the combining of the classifiers within each cluster leads to better performance compared to

combining all classifiers to produce a global model or using individual classifiers at each site.

FUTURE TRENDS

One trend that can be noticed during the last years is the implementation of DDM systems using emerging distributed computing paradigms such as Web services and the application of DDM algorithms in emerging distributed environments, such as mobile networks, sensor networks, grids and peer-to-peer networks.

Cannataro and Talia (2003), introduced a reference software architecture for knowledge discovery on top of computational grids, called *Knowledge Grid*. Datta, Bhaduri, Giannela, Kargupta and Wolff (2006), present an overview of DDM applications and algorithms for P2P environments. McConnell and Skillicorn (2005) present a distributed approach for prediction in sensor networks, while Davidson and Ravi (2005) present a distributed approach for data pre-processing in sensor networks.

CONCLUSION

DDM enables learning over huge volumes of data that are situated at different geographical locations. It supports several interesting applications, ranging from fraud and intrusion detection, to market basket analysis over a wide area, to knowledge discovery from remote sensing data around the globe.

As the network is increasingly becoming the computer, the role of DDM algorithms and systems will continue to play an important role. New distributed applications will arise in the near future and DDM will be challenged to provide robust analytics solutions for these applications.

REFERENCES

Agrawal, R. & Shafer J.C. (1996). Parallel Mining of Association Rules. *IEEE Transactions on Knowledge and Data Engineering, 8*(6), 962-969.

Agrawal R. & Srikant, R. (1994, September). *Fast Algorithms for Mining Association Rules*. In Proceedings of the 20th International Conference on Very Large Databases (VLDB'94), Santiago, Chile, 487-499.

Ashrafi, M. Z., Taniar, D. & Smith, K. (2004). ODAM: An Optimized Distributed Association Rule Mining Algorithm. *IEEE Distributed Systems Online, 5*(3).

Cannataro, M. and Talia, D. (2003). The Knowledge Grid. *Communications of the ACM, 46*(1), 89-93.

Chan, P. & Stolfo, S. (1993). *Toward parallel and distributed learning by meta-learning*. In Proceedings of AAAI Workshop on Knowledge Discovery in Databases, 227-240.

Cheung, D.W., Han, J., Ng, V., Fu, A.W. & Fu, Y. (1996, December). *A Fast Distributed Algorithm for Mining Association Rules*. In Proceedings of the 4th International Conference on Parallel and Distributed Information System (PDIS-96), Miami Beach, Florida, USA, 31-42.

Cheung, D.W., Ng, V., Fu, A.W. & Fu, Y. (1996). Efficient Mining of Association Rules in Distributed Databases. *IEEE Transactions on Knowledge and Data Engineering, 8*(6), 911-922.

Cho, V. & Wüthrich, B. (2002). Distributed Mining of Classification Rules. *Knowledge and Information Systems, 4*, 1-30.

Datta, S, Bhaduri, K., Giannella, C., Wolff, R. & Kargupta, H. (2006). Distributed Data Mining in Peer-to-Peer Networks, *IEEE Internet Computing 10*(4), 18-26.

Davidson I. & Ravi A. (2005). *Distributed Pre-Processing of Data on Networks of Berkeley Motes Using Non-Parametric EM*. In Proceedings of 1st International Workshop on Data Mining in Sensor Networks, 17-27.

Fan, W., Stolfo, S. & Zhang, J. (1999, August). *The Application of AdaBoost for Distributed, Scalable and On-Line Learning*. In Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, California, USA, 362-366.

Fu, Y. (2001). Distributed Data Mining: An Overview. *Newsletter of the IEEE Technical Committee on Distributed Processing*, Spring 2001, pp.5-9.

Guo, Y. & Sutiwaraphun, J. (1999). *Probing Knowledge in Distributed Data Mining*. In Proceedings of

the Third Pacific-Asia Conference on Methodologies for Knowledge Discovery and Data Mining (PAKDD-99), 443-452.

Hall, L.O., Chawla, N., Bowyer, K. & Kegelmeyer, W.P. (2000). Learning Rules from Distributed Data. In M. Zaki & C. Ho (Eds.), *Large-Scale Parallel Data Mining*. (pp. 211-220). LNCS 1759, Springer.

Hall, L.O., Chawla, N. & Bowyer, K. (1998, July). *Decision Tree Learning on Very Large Data Sets*. In Proceedings of the IEEE Conference on Systems, Man and Cybernetics.

Hall, L.O., Chawla, N. & Bowyer, K. (1997). *Combining Decision Trees Learned in Parallel*. In Proceedings of the Workshop on Distributed Data Mining of the 4th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

Johnson, E.L & Kargupta, H. (1999). Collective Hierarchical Clustering from Distributed, Heterogeneous Data. In M. Zaki & C. Ho (Eds.), *Large-Scale Parallel Data Mining*. (pp. 221-244). LNCS 1759, Springer.

Kargupta, H., Park, B-H, Herschberger, D., Johnson, E. (2000) Collective Data Mining: A New Perspective Toward Distributed Data Mining. In H. Kargupta & P. Chan (Eds.), *Advances in Distributed and Parallel Knowledge Discovery*. (pp. 133-184). AAAI Press.

Lazarevic, A, & Obradovic, Z. (2001, August). The Distributed Boosting Algorithm. In Proceedings of the 7th ACM SIGKDD international conference on Knowledge discovery and data mining, San Francisco, California, USA, 311-316.

McClean, S., Scotney, B., Greer, K. & P Páircéir, R. (2001). *Conceptual Clustering of Heterogeneous Distributed Databases*. In Proceedings of the PKDD'01 Workshop on Ubiquitous Data Mining.

McConnell S. and Skillicorn D. (2005). *A Distributed Approach for Prediction in Sensor Networks*. In Proceedings of the 1st International Workshop on Data Mining in Sensor Networks, 28-37.

Park, B. & Kargupta, H. (2003). Distributed Data Mining: Algorithms, Systems, and Applications. In N. Ye (Ed.), *The Handbook of Data Mining*. (pp. 341-358). Lawrence Erlbaum Associates.

Parthasarathy, S. & Ogihara, M. (2000). *Clustering Distributed Homogeneous Databases*. In Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD-00), Lyon, France, September 13-16, 566-574.

Samatova, N.F., Ostrouchov, G., Geist, A. & Melechko A.V. (2002). RACHET: An Efficient Cover-Based Merging of Clustering Hierarchies from Distributed Datasets. *Distributed and Parallel Databases 11*, 157-180.

Schapire, R & Singer, Y. (1999). Improved boosting algorithms using confidence-rated predictions. *Machine Learning 37*(3), 297-336.

Tsoumakas, G., Angelis, L. & Vlahavas, I. (2003). Clustering Classifiers for Knowledge Discovery from Physically Distributed Databases. *Data & Knowledge Engineering 49*(3), 223-242.

Wolpert, D. (1992). Stacked Generalization. *Neural Networks 5*, 241-259.

Wüthrich, B. (1997). Discovery probabilistic decision rules. International Journal of Information Systems in Accounting, Finance, and Management 6, 269-277.

KEY TERMS

Data Skewness: The observation that the probability distribution of the same attributes in distributed databases is often very different.

Distributed Data Mining (DDM): A research area that is concerned with the development of efficient algorithms and systems for knowledge discovery in distributed computing environments.

Global Mining: The combination of the local models and/or sufficient statistics in order to produce the global model that corresponds to all distributed data.

Grid: A network of computer systems that share resources in order to provide a high performance computing platform**.**

Homogeneous and Heterogeneous Databases: The schemata of Homogeneous (Heterogeneous) databases contain the same (different) attributes.

Local Mining: The application of data mining algorithms at the local data of each distributed site.

Sensor Network: A network of spatially distributed devices that use sensors in order to monitor environment conditions.

Document Indexing Techniques for Text Mining

José Ignacio Serrano
Instituto de Automática Industrial (CSIC), Spain

Mª Dolore del Castillo
Instituto de Automática Industrial (CSIC), Spain

INTRODUCTION

Owing to the growing amount of digital information stored in natural language, systems that automatically process text are of crucial importance and extremely useful. There is currently a considerable amount of research work (Sebastiani, 2002; Crammer et al., 2003) using a large variety of machine learning algorithms and other Knowledge Discovery in Databases (KDD) methods that are applied to Text Categorization (automatically labeling of texts according to category), Information Retrieval (retrieval of texts similar to a given cue), Information Extraction (identification of pieces of text that contains certain meanings), and Question/Answering (automatic answering of user questions about a certain topic). The texts or documents used can be stored either in ad hoc databases or in the World Wide Web. Data mining in texts, the well-known Text Mining, is a case of KDD with some particular issues: on one hand, the features are obtained from the words contained in texts or are the words themselves. Therefore, text mining systems faces with a huge amount of attributes. On the other hand, the features are highly correlated to form meanings, so it is necessary to take the relationships among words into account, what implies the consideration of syntax and semantics as human beings do. KDD techniques require input texts to be represented as a set of attributes in order to deal with them. The text-to-representation process is called text or document indexing, and the attributes and called indexes. Accordingly, indexing is a crucial process in text mining because indexed representations must collect, only with a set of indexes, most of the information expressed in natural language in the texts with the minimum loss of semantics, in order to perform as well as possible.

BACKGROUND

The traditional "bag-of-words" representation (Sebastiani, 2002) has shown that a statistical distribution of word frequencies, in many text classification problems, is sufficient to achieve high performance results. However, in situations where the available training data is limited by size or by quality, as is frequently true in real-life applications, the mining performance decreases. Moreover, this traditional representation does not take into account the relationships among the words in the texts so that if the data mining task required abstract information, the traditional representation would not afford it. This is the case of the textual informal information in web pages and emails, which demands a higher level of abstraction and semantic depth to perform successfully.

In the end-nineties, word hyperspaces appeared on the scene and they are still updating and improving nowadays. These kind of systems build a representation, a matrix, of the linguistic knowledge contained in a given text collection. They are called word hyperspaces because words are represented in a space of a high number of dimensions. The representation, or hyperspace, takes into account the relationship between words and the syntactic and semantic context where they occur and store this information within the knowledge matrix. This is the main difference with the common "bag of words" representation. However, once the hyperspace has been built, word hyperspace systems represent the text as a vector with a size equal to the size of the hyperspace by using the information hidden in it, and by doing operations with the rows and the columns of the matrix corresponding to the words in the texts.

LSA (Latent Semantic Analysis) (Landauer, Foltz & Laham, 1998; Lemaire & Denhière, 2003) was the first

one to appear. Given a text collection, LSA constructs a term-by-document matrix. The A_{ij} matrix component is a value that represents the relative occurrence level of term *i* in document *j*. Then, a dimension reduction process is applied to the matrix, concretely the SVD (Singular Value Decomposition) (Landauer, Foltz & Laham, 1998). This dimension-reduced matrix is the final linguistic knowledge representation and each word is represented by its corresponding matrix row of values (vector). After the dimension reduction, the matrix values contain the latent semantic of all the other words contained in all each document. A text is then represented as a weighted average of all the vectors corresponding to the words it contains and the similarity between two texts is given by the cosine distance between the vectors that represent them.

Hyperspace Analogue to Language (HAL) (Burgess, 2000) followed LSA. In this method, a matrix that represents the linguistic knowledge of a text collection is also built but, in this case, is a word-by-word matrix. The A_{ij} component of the matrix is a value related to the number of times the word *i* and the word *j* co-occur within the same context. The context is defined by a window of words, of a fixed size. The matrix is built by sliding the window over all the text in the collection, and by updating the values depending on the distance, in terms of position, between each pair of words in the window. A word is represented by the values corresponding to its row concatenated with the values corresponding to its column. This way, not only the information about how is the word related to each other is considered, but also about how the other words are related to it. The meaning of a word can be derived from the degrees of the relations of the word with each other. Texts are also represented by the average of the vectors of the words it contains and compared by using the cosine distance.

Random Indexing (Kanerva, Kristofersson & Holst, 2000) also constructs a knowledge matrix but in a distributed fashion and with a strong random factor. A fixed number of contexts (mainly documents) in which words can occur, is defined. Each context is represented by a different random vector of a certain size. The vector size is defined by hand and corresponds to the number of columns, or dimensions, of the matrix. Each row of the matrix makes reference to one of the words contained in the text collection from which the linguistic knowledge was obtained. This way, each time a word occurs in one of the predefined contexts, the context vector is summed up to the row referent to the word. At the end of the construction of the knowledge matrix, each word is represented as a vector resulting from the sum of all the vectors of the contexts where it appears. Other advantage relies on the flexibility of the model, because the incorporation of a new context or word only implies a new random vector and a sum operation. Once again, texts are represented as the average (or any other statistical or mathematical function) of the vector of the words that appear in it.

Unlike the previous systems, in WAS (Word Association Space) (Steyvers, Shiffrin & Nelson, 2004) the source of the linguistic knowledge is not a text collection but data coming from human subjects. Although the associations among words are represented as a word-by-word matrix, they are not extracted from the co-occurrences within texts. The association norms are directly queried humans. A set of human subjects were asked to write the first word that come out in their mind when each of the words in a list were presented to them, one by one, so that the given words correspond to the rows of the matrix and the answered words correspond to the columns of the matrix. Finally, the SVD dimension reduction is applied to the matrix. The word and text representations are obtained the same way as the systems above.

The FUSS (Featural and Unitary Semantic Space) system (Vigliocco, Vinson, Lewis & Garrett, 2004), the knowledge source also comes from human subjects. It is based on the state that words are not only associated to their semantic meaning but also to the way humans learn them when perceive them. Then, human subjects are asked to choose which conceptual features are useful to describe each entry of a word list. So a word-by-conceptual feature matrix is constructed, keeping the *k* most considered features and discarding the others, and keeping the *n* most described words by the selected features. Once the matrix is bounded, a SOM (Self Organizing Map) algorithm is applied. A word is represented by the most activated unit of the maps, when the feature vector which corresponds to the word is taken as the input of the map. The texts are then represented by the most activated units which correspond to the words that appear inside it.

In Sense Clusters system (Pedersen & Kulkarni, 2005), the authors propose two different representations for the linguistic knowledge, both of them matrix-like, called representation of first order and second order, respectively. In the first representations, matrix values

correspond to the frequency of occurring for the words, bigrams and n-grams represented by the columns, in the contexts of the rows. In the other representation, a square matrix collects the associations among the words that belong to all the bigrams and n-grams. The SVD dimension reduction method is applied to both matrices and then a clustering algorithm is run over them. This way, the semantics topics are described by the clusters obtained. A new text is represented as a vector separately for each cluster. Finally, the decisions are taken following the similarity criterion of the text to each cluster.

The system that stands for NLS (Non-Latent Similarity) (Cai, McNamara, Louwerse, Hu, Rowe & Graesser, 2004) is one of the few that deals with syntax in an explicit way. A set of features is defined by an expert, as well as a set of significance weights for the features. The features are sequences of POS (Part Of Speech) tags. For each word in the text collection, a vector of size equal to the number of defined features is created. Then a word-by-word matrix is built. An SVD dimension reduction is finally applied to the matrix. Texts are represented as a function of the vectors of the words it contains.

Besides the systems briefly described above, there exist other techniques, also different from the traditional indexing, that do not use a matrix to collect the background linguistic knowledge. Instead of that, PMI (Pointwise Mutual Information) (Turney, 2001), the most representative example of this kind of systems, represents the word associations as a probabilistic model. The frequencies of co-occurrence are obtained from the World Wide Web, by asking the Altavista Advanced Search, so that if a word is not in the probabilistic model, the co-occurrence relationships with the other words in the model are obtained in real time from web search results, opposite to the classical statistical models which fail when missing words appear.

The CDSA (Context Dependent Sentence Abstraction) model (Ventura, Hu, Graesser, Louwerse & Olney, 2004) does not use a matrix either. For each word, a list of the words with which it co-occurrs ("neighbours") is stored together with a weight for each component of the list. The weights represent the frequencies of co-occurrence. This way, the similarity between two words is a function on the weights of their shared neighbours.

MAIN FOCUS

All the systems described above improve the traditional "bag of words" representation. They have provided great advances to text mining in terms of depth and richness for text representation. In spite of the progress made with word hyperspaces and probabilistic models, human beings keep doing text mining tasks much better than machines, although of course more slowly. So, in order to certainly obtain good results a brining near human cognition is required. However, it is hard to believe that linguistic knowledge is represented as a matrix in the human mind and that text reading is carried out by mathematical operations on this matrix. Although many of the systems try to be close to humans, by searching the psychological plausibility for some of the internal processes, they do not achieve it in fact. Let us put attention in some of the features that characterize a computational model of linguistic representation (Lemaire & Denhière, 2004):

1. **Input:** The source from which the linguistic knowledge is constructed. A corpus of heterogeneous documents, mixing different topics, styles and spoken and writing texts, is psychologically more plausible than small sets of association norms or sublanguages.
2. **Representation:** The formalism to store and operate with words and texts. Vectors allow to easily compute similarity among words and texts. However, it makes difficult to find out the most related words or texts to a given one. Moreover, the asymmetric condition of the similarity function among vectors is not psychologically plausible, and neither the use of the same representation for vectors and words.
3. **Knowledge updating:** The way in which the linguistic knowledge can be modified when new texts need to be added. The incremental models are close to humans. In some of the methods described above, updating is very time-consuming because it implies the reconstruction of all the knowledge representation from the beginning.
4. **High-order co-occurrences:** The indirect associations or associations of the words that are associated to other and so on. Only a few of the models take several levels of indirect associations into account, what is showed to be very significant

in human language cognition.

5. **Compositionality:** The process to represent the text from the words it contains. A direct composition, such as vector average or sum, is very efficient in computational terms. However, humans do not create the representation of a text in mind by a direct function on the words of the texts.

There is only one system that introduces a structural representation: ICAN (Incremental Construction of an Associative Network) (Lemaire & Denhière, 2004). ICAN does not store linguistic knowledge as a matrix but as a net of associated words. These associations have a weight calculated from probabilities of co-occurrence and non-co-occurrence between word pairs. In this model words and texts do not share the same structure of representation, unlike the systems mentioned above. This model makes it possible to incrementally add new words without retraining and recalculating the knowledge, which is psychologically more plausible. This approach proposes a connectionist model, by representing linguistic knowledge as a net of concepts associated by context. However, texts are simply represented by subnets of the global knowledge net, formed by the nodes corresponding to the words in the texts and their associations.

However, humans represent texts in mind after an indirect process: reading. Human reading is a process of sequential perception over time, during which the mind builds mental images and inferences which are reinforced, updated or discarded until the end of the text (Perfetti, 1999). At that moment, this mental image allows humans to summarize and classify the text, to retrieve similar texts or simply to talk about the text by expressing opinions. Thus, the target of the research should be the construction of text representations as a result of a process over time, with a structure that makes it possible to indirectly describe the salience and relations of words at every instant during the reading process. There are some research works referring to computational models of reading (Ram & Moorman, 1999) in terms of human cognition. However, these systems are used in a closed linguistic environment with a very limited sublanguage and very specific targets. They are theoretical models rather than applications to real data mining problems.

FUTURE TRENDS

Future research is intended to bring together different research fields, such as cognitive science, cognitive linguistics, cognitive neuroscience, computational linguistics, artificial intelligence and machine learning among others. Word hyperspaces and probabilistic systems for indexing documents will be given of psychological plausibility so that they approximate to human cognition. Hybrid systems will be developed, which will make use of the models of cognitive theories but keeping the capacity of hyperspaces and probabilistic models to deal with data-intensive real problems. With this leaning to human cognition, the techniques for representing documents will intend to also model individual user behavior in order to obtain customized representations and to optimize the performance of text mining tasks.

CONCLUSION

On one hand, word hyperspaces and probabilistic systems have been shown to highly improve traditional "bag-of-words" representation of texts, what implies an increasing of the performance of text mining systems. Although they include information about syntax and semantics and they are able to deal with a huge amount of texts, they still produce representations as vectors without structure, in a direct fashion. Humans represent texts in mind as a result of a process of reading over time. On the other hand, models of cognitive reading deal with these mental processes but in closed environments and oriented to very specific tasks. Hybrid approaches that bring together hyperspaces and cognition, as well as the real world application and the theory, will be the future systems for indexing documents in order to optimize text mining tasks by user-customized representations.

REFERENCES

Burgess, C. (1998). From simple associations to the building blocks of language: Modeling meaning in memory with the HAL model, *Behavior Research*

Methods, Instruments, & Computers, 30, 188-198.

Cai, Z., McNamara, D.S., Louwerse, M., Hu, X., Rowe, M. & Graesser, A.C. (2004). NLS: A non-latent similarity algorithm, *In Proceedings of the 26th Annual Meeting of the Cognitive Science Society (CogSci'2004)*, 180-185.

Crammer, K., Kandola, J. & Singer Y. (2003). Online Classification on a Budget, *Advances in Neural Information Processing Systems*, 16, Thrun, Saul and Scholkopf editors, MIT Press, Cambridge.

Kanerva, P., Kristofersson, J. & Holst, A. (2000). Random indexing of text samples for Latent Semantic Analysis, *Proceedings of the 22nd Annual Conference of the Cognitive Science Society* , 1036-.

Landauer, T. K., Foltz, P. W. & Laham, D. (1998). An introduction to Latent Semantic Analysis, *Discourse Processes*, 25, 259-284.

Lemaire, B. & Denhière, G. (2003). Cognitive models based on Latent Semantic Analysis, *Tutorial given at the 5th International Conference on Cognitive Modeling (ICCM'2003)*, Bamberg, Germany, April 9.

Lemaire, B. & Denhière, G. (2004). Incremental construction of an associative network from a corpus, *In Proceedings of the 26th Annual Meeting of the Cognitive Science Society (CogSci'2004)*, 825-830.

Pedersen, T. & Kulkarni, A. (2005). Identifying similar words and contexts in natural language with SenseClusters, *In Proceedings of the Twentieth National Conference on Artificial Intelligence (Intelligent Systems Demonstration)*.

Perfetti, C. A. (1999). Comprehending written language: A blue print of the reader, *The Neurocognition of Language*, Brown & Hagoort Eds., Oxford University Press, 167-208.

Ram A. & Moorman K. (1999). *Understanding Language Understanding: Computational Models of Reading*, Ashwin Ram & Kenneth Moorman (eds.), Cambridge: MIT Press.

Sebastiani, F. (2002). Machine learning in automated text categorization, *ACM Computing Surveys*, 34(1), 1-47.

Steyvers, M., Shiffrin R.M., & Nelson, D.L. (2004). Word association spaces for predicting semantic similarity effects in episodic memory, *In A. Healy (Ed.), Experimental Cognitive Psychology and its Applications: Festschrift in Honor of Lyle Bourne, Walter Kintsch, and Thomas Landauer,* Washington DC: American Psychological Association.

Turney, P. (2001). Mining the Web for synonyms: PMI-IR versus LSA on TOEFL, *In De Raedt, Luc and Flach, Peter, Eds., Proceedings of the Twelfth European Conference on Machine Learning (ECML-2001)* , 491-502.

Ventura, M., Hu, X., Graesser, A., Louwerse, M. & Olney, A. (2004). The context dependent sentence abstraction model, *In Proceedings of the 26th Annual Meeting of the Cognitive Science Society (CogSci'2004)*, 1387-1392.

Vigliocco, G., Vinson, D.P, Lewis, W. & Garrett, M.F. (2004). Representing the meanings of object and action words: The Featural and Unitary Semantic System (FUSS) hypothesis, *Cognitive Psychology*, 48, 422-488.

KEY TERMS

Bag of Words: Traditional representation of documents stored electronically in which texts are represented by the list of the words they contains together with some measurement of the local and global frequency of the words.

Context: Continuous fragment or unit of text in which the words that constitute it are considered to be semantically or syntactically associated to each other in some manner.

Document Indexing: Representation of texts stored electronically by a set of features called indexes, mainly words or sequences of words, which try to collect most of the information contained in the natural language of the text, in order to be dealt by text mining methods.

Information Retrieval: Process of automatic searching, collecting and ranking a set of texts or fragments of text in natural language from a corpus, semantically related to an input query stated by the user also in natural language.

Latent Semantic: Relationship among word meanings given by the context in which words co-occur and

by the indirect relationships between the words.

Text Classification: Process of automatic representation and labeling of texts that intends to characterize the documents as belonging, in some degree, to one or more predefined thematic categories.

Text Mining: Knowledge discovery process on data given as text in natural language by the analysis of the textual information in many different ways. It is a kind of data mining which deals with a huge amount of a special type of information where preprocessing and feature selection steps are of crucial importance with this type of data. It is also known as text data mining, knowledge discovery in text or intelligent text analysis.

Word Hyperspace: Mathematical space with a high number of dimensions in which words are unequivocally compared and represented as vectors with a number of components equal to the number of dimensions. Word hyperspaces are usually constructed automatically from examples of natural language texts by statistical and mathematical methods.

Dynamic Data Mining

Richard Weber
University of Chile, Chile

INTRODUCTION

Since the First KDD Workshop back in 1989 when "Knowledge Mining" was recognized as one of the top 5 topics in future database research (Piatetsky-Shapiro 1991), many scientists as well as users in industry and public organizations have considered data mining as highly relevant for their respective professional activities.

We have witnessed the development of advanced data mining techniques as well as the successful implementation of knowledge discovery systems in many companies and organizations worldwide. Most of these implementations are static in the sense that they do not contemplate explicitly a changing environment. However, since most analyzed phenomena change over time, the respective systems should be adapted to the new environment in order to provide useful and reliable analyses.

If we consider for example a system for credit card fraud detection, we may want to segment our customers, process stream data generated by their transactions, and finally classify them according to their fraud probability where fraud pattern change over time. If our segmentation should group together homogeneous customers using not only their current feature values but also their trajectories, things get even more difficult since we have to cluster vectors of functions instead of vectors of real values. An example for such a trajectory could be the development of our customers' number of transactions over the past six months or so if such a development tells us more about their behavior than just a single value; e.g., the most recent number of transactions.

It is in this kind of applications is where dynamic data mining comes into play!

Since data mining is just one step of the iterative KDD (Knowledge Discovery in Databases) process (Han & Kamber, 2001), dynamic elements should be considered also during the other steps. The entire process consists basically of activities that are performed before doing data mining (such as: selection, pre-processing, transformation of data (Famili et al., 1997)), the actual data mining part, and subsequent steps (such as: interpretation, evaluation of results).

In subsequent sections we will present the background regarding dynamic data mining by studying existing methodological approaches as well as already performed applications and even patents and tools. Then we will provide the main focus of this chapter by presenting dynamic approaches for each step of the KDD process. Some methodological aspects regarding dynamic data mining will be presented in more detail. After envisioning future trends regarding dynamic data mining we will conclude this chapter.

BACKGROUND

In the past a diverse terminology has been used for emerging approaches dealing with "dynamic" elements in data mining applications. Learning from data has been defined as *incremental* if the training examples used become available over time, usually one at a time; see e.g., (Giraud-Carrier, 2000). Mining *temporal* data deals with the analysis of streams of categorical data (e.g., events; see e.g., Domingos, Hulten, 2003) or the analysis of time series of numerical data (Antunes, Oliveira 2001; Huang, 2007). Once a model has been built, *model updating* becomes relevant. According to the CRISP-DM methodology such updating is part of the monitoring and maintenance plan to be performed after model construction.

The following listing provides an overview on applications of dynamic data mining.

- Intrusion detection (Caulkins et al., 2005).
- Traffic state identification (Crespo, Weber, 2005).
- Predictive maintenance (Joentgen et al., 1999).
- Scenario analysis (Weber 2007).
- Time series prediction (Kasabov, Song, 2002)

Dynamic data mining has also been patented already, e.g., the dynamic improvement of search engines in internet which use so-called rating functions in order to measure the relevance of search terms. "Based upon a historical profile of search successes and failures as well as demographic/personal data, technologies from artificial intelligence and other fields will optimize the relevance rating function. The more the tool is used (especially by a particular user) the better it will function at obtaining the desired information earlier in a search. ... The user will just be aware that with the same input the user might give a static search engine, the present invention finds more relevant, more recent and more thorough results than any other search engines." (Vanderveldt, Black 2001).

MAIN FOCUS

As has been shown above dynamic data mining can be seen as an area within data mining where dynamic elements are considered. This can take place in any of the steps of the KDD process as will be introduced next.

Feature Selection in Dynamic Data Mining

Feature selection is an important issue of the KDD process before the actual data mining methods are applied. It consists in determining the most relevant features given a set of training examples (Famili et al., 1997). If, however, this set changes dynamically over time the selected feature set could do so as well. In such cases we would need a methodology that helps us to dynamically update the set of selected features. A dynamic wrapper approach for feature selection has been proposed in (Guajardo et al., 2006) where feature selection and model construction is performed simultaneously.

Preprocessing in Dynamic Data Mining

If in certain applications feature trajectories instead of feature values are relevant for our analysis we are faced with specific requirements for data preprocessing. In such cases it could be necessary to determine distances between trajectories within the respective data mining algorithms (as e.g., in Weber, 2007) or alternatively to apply certain preprocessing steps in order to reduce the trajectories to real-valued feature vectors.

Dynamic Clustering

Clustering techniques are used for data mining if the task is to group similar objects in the same classes (segments) whereas objects from different classes should show different characteristics (Beringer, Hüllermeier 2007). Such clustering approaches could be generalized in order to treat different dynamic elements, such as e.g., dynamic objects and/or dynamic classes. Clustering of **dynamic objects** could be the case where trajectories of feature vectors are used as input for the respective data mining algorithms. If the class structure changes over time we speak about **dynamic classes**. We present first approaches for clustering of dynamic objects and then a methodology to determine dynamic classes.

In order to be able to cluster dynamic objects, we need a distance measure between two vectors where each component is a trajectory (function) instead of a real number. Functional fuzzy c-means (FFCM) is a fuzzy clustering algorithm where the respective distance is based on the similarity between two trajectories which is determined using membership functions (Joentgen et al., 1999).

Applying this distance measure between functions FFCM determines classes of dynamic objects. The respective class centers are composed of the most representative trajectories in each class; see the following figure.

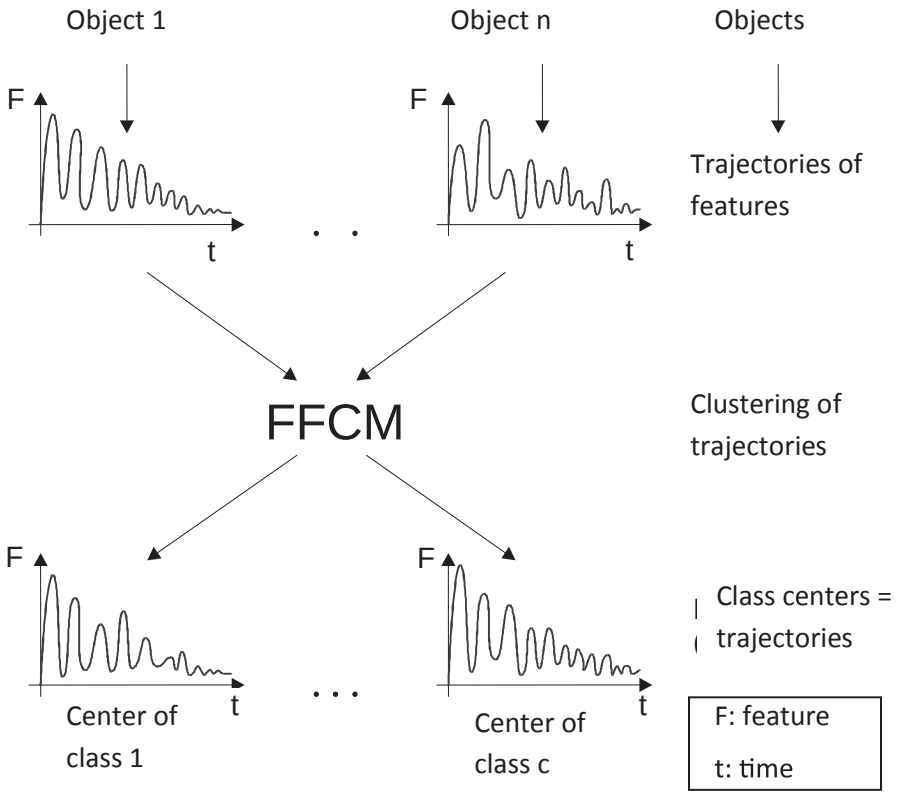
Determine dynamic classes could be the case when classes have to be created, eliminated or simply moved in the feature space. The respective methodology applies the following five steps in order to detect these changes.

Here we present just the methodology's main ideas; a detailed description can be found e.g., in (Crespo, Weber, 2005). It starts with a given classifier; in our case we chose Fuzzy c-means since the respective membership values provide a strong tool for classifier updating.

Step I: Identify Objects that Represent Changes

For each new object we want to know if it can be explained well by the given classifier. With other words

Figure 1. Functional fuzzy c-means (FFCM)



we want to identify objects that represent possible changes in the classifier structure because they are not well classified. If there are many objects that represent such possible changes we proceed with Step II, in the other case we go immediately to Step III.

Step II: Determine Changes of Class Structure

In Step I we have identified the objects that could not be well classified. Now we want to decide if we have to change the classifier structure (i.e. create new classes) or if it is sufficient to just move the existing classes. If "many new objects" are identified in Step I to represent changes we have to create a new class, in the other case we just move the existing classes in the feature space.

Step III: Change the Class Structure

Here we perform the changes according to Steps I and II (move or create classes).

Step III.1 Move Classes

We update the position of the existing class centers having the new objects and knowing that they do not ask for a structural change.

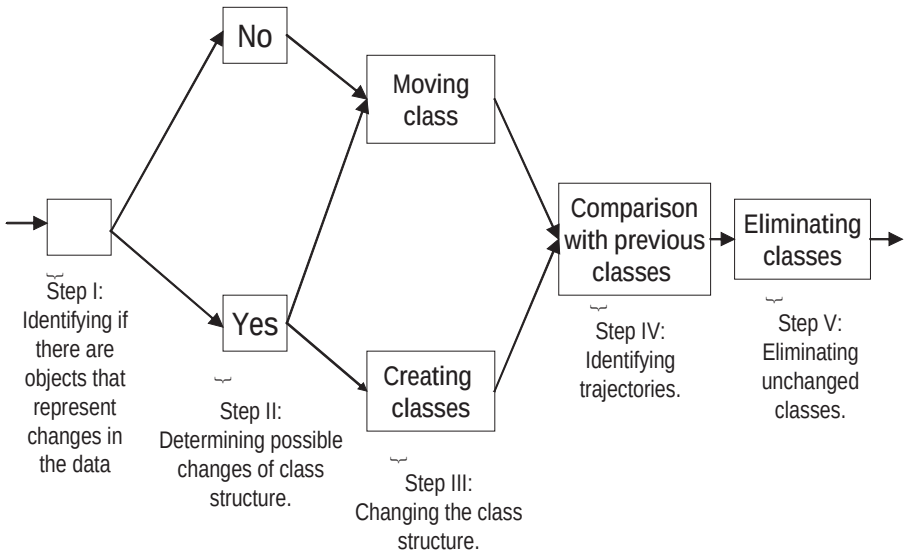
Step III.2 Create Classes

If we know that classes have to be created, we first determine an appropriate class number. Then we apply fuzzy c-means with the new class number to the available data set.

Step IV: Identify Trajectories of Classes

We identify trajectories of the classes from the previous cycles in order to decide if they received new objects. Classes that did not receive new objects during several cycles have to be eliminated.

Figure 2. Methodology for dynamic clustering



Step V: Eliminate Unchanged Classes

Based on the result of Step IV we eliminate classes that did not receive new objects during an "acceptable period".

Figure 2 provides a general view of the proposed methodology.

Dynamic Regression and Classification

Regression or classification models have to be updated over time when new training examples represent different pattern. A methodology proposed for updating forecasting models based on Support Vector Regression (SVR) changes constantly the composition of the data sets used for training, validation, and test (Weber, Guajardo, 2008). The basic idea is to add always the most recent examples to the training set in order to extract this way the most recent information about pattern changes. We suppose the series to be characterized by seasonal patterns (cycles) of length *c*.

In the static case we sequentially split data into training, validation and test sets whereas in the dynamic approach these sets are changed periodically as shown in the following figure.

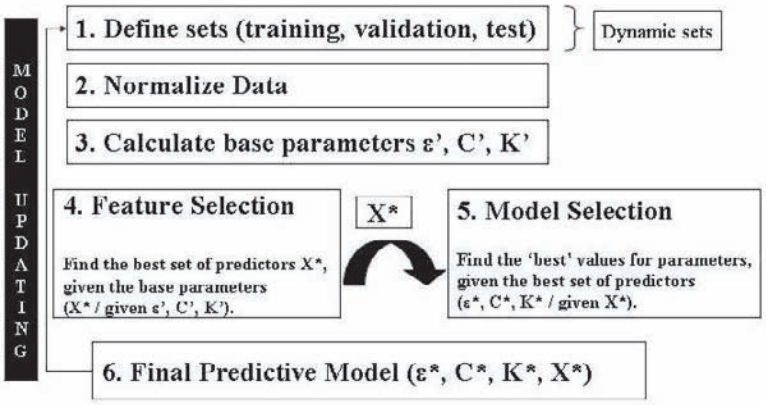
The proposed model updating strategy is designed to deal with time series with seasonal patterns. We will refer to a complete seasonal pattern as a cycle; examples are e.g., monthly data with yearly seasonality, where a cycle is defined by a year.

First, we have to identify the length of the cycles of the series. For this purpose, graphical inspection, autocorrelation analysis or spectral analysis can be carried out. Then we define a test set containing at least two cycles. This test set is divided into subsets, each one containing a single cycle. Let there be k cycles of length c belonging to the test set in the static case. In the static case, we construct just one model to predict all the observations contained in the test set, as well as for future observations, maintaining this model throughout the entire procedure unchanged. The main idea of our updating approach consists of developing different predictive models for each cycle of the test set, as well as for any future cycle by using the most recent information for model construction.

Next, we have to define the configuration of the training and validation sets for predicting the first cycle of the test set. Training data contains two parts where the first part is a set of past (or historical) data and the second part contains the most recent information. The idea of integrating the most recent information into the training set is to obtain more accurate estimations by taking into account the most recent patterns in model construction, while the proportion of data belonging to training and validation is kept stable over time.

To predict the second cycle of the test set, we add the first cycle of the test set (this data is now part of the

Figure 3. Methodology for updating regression-based forecasting using SVR



past) to the training set, and build a different predictive model. By doing so, we ensure again that most recent information has influence in model construction.

As we want to keep the proportion of data belonging to the training and validation sets stable over time, we move data from the historical part of the training set to the validation set. The amount of data to be moved is determined according to the original proportion of data in each subset. This same strategy will be applied to build predictive models for the rest of the cycles of the test set, as well as for any future observations.

Approaches for updating classification models have been presented e.g., in (Aggarwal et al., 2005, Aggarwal, 2007) where a combination of clustering and classification detects changes in stream data.

FUTURE TRENDS

In the future more *methods* for dynamic data mining addressing different steps of the KDD process will be developed. We will also see many interesting *applications* since most phenomena in real-world applications possess inherently dynamic elements.

Consequently, tool providers are supposed to add such modules to their *commercial products* in order to assist their customers more efficiently. A first though simple tool exists already on the market: the CLEO module from SPSS that deploys the entire KDD process for dynamic data mining that can be quickly adapted to changes in the analyzed phenomena. In the case of data stream mining it would also be interesting to develop special *hardware tools* in order to optimize processing speed of the respective mining tasks.

At the annual KDD conferences a particular workshop dedicated to *standards* for data mining has been established and reports since 2003 about maturing standards, including the following:

- Predictive Model Markup Language (PMML)
- XML for Analysis and OLE DB for Data Mining
- SQL/MM Part 6: Data Mining
- Java Data Mining (JDM)
- CRoss Industry Standard Process for Data Mining (CRISP-DM)
- OMG Common Warehouse Metadata (CWM) for Data Mining

The rising need for dynamic data mining will lead also to standards regarding the respective models in order to be "transferable" between tools from different providers and "usable" for various users.

A recent theoretical development which promises a huge potential is the *combination of dynamic data mining with game theory*. While data mining analyzes the behavior of agents in a real-world context via an analysis of data they generated, game theory tries to explain theoretically the behavior of such agents. If these agents "play" several times their game it will not be sufficient to just apply static data mining. Dynamic data mining will reveal the specific patterns in a changing environment. First experiments in this direction have shown already very promising results (Bravo, Weber, 2007).

CONCLUSION

Dynamic data mining will become one of the key elements of future knowledge discovery applications. Phenomena in real-world applications are dynamic in nature and so should be the models that we apply to extract useful information. This chapter presented an overview on recent developments as well as future trends related to this area.

ACKNOWLEDGMENT

The work presented in this chapter has been financed by the Chilean Fondecyt Project 1040926 and the Scientific Millennium Institute "Sistemas Complejos de Ingeniería" (www.sistemasdeingenieria.cl).

REFERENCES

Aggarwal, Ch. C., Han, J., Wang, J., & Yu, P. S. (2005): On High Dimensional Projected Clustering of Data Stream. *Data Mining and Knowledge Discovery 10*, (3), 251-273

Aggarwal, Ch. C. (Ed.) (2007). *Data Streams – Models and Algorithms*. Springer.

Antunes, C. M., & Oliveira, A. L. (2001), *Temporal Data Mining: an overview.* Workshop on Temporal Data Mining, (KDD2001). San Francisco, September 2001, 1-13

Beringer, J., & Hüllermeier, E. (2007). Fuzzy Clustering of Parallel Data Streams. In J. Valente de Oliveira, and W. Pedrycz (Eds.), *Advances in Fuzzy Clustering and its Applications* (pp. 333-352). John Wiley and Sons.

Bravo, C., & Weber, R. (2007). *Modelo de Tarificación en Base a SVMs y Teoría de Juegos*. In: Óptima 2007, Puerto Montt, Chile (in Spanish)

Caulkins, B.D., Lee, J., & Wang, M. (2005). *A Dynamic Data Mining Technique for Intrusion Detection Systems*. 43rd ACM Southeast Conference, March 18-20, 2005, Kennesaw, GA

Crespo, F., & Weber, R. (2005). A Methodology for Dynamic Data Mining based on Fuzzy Clustering. *Fuzzy Sets and Systems, 150*(2), 267-284

Domingos, P., & Hulten, G. (2003). A General Framework for Mining Massive Data Streams. *Journal of Computational and Graphical Statistics 12*(4), 945-949.

Famili, A., Shen, W.-M., Weber, R., & Simoudis, E. (1997). Data Preprocessing and Intelligent Data Analysis. *Intelligent Data Analysis 1*(1), 3-23.

Giraud-Carrier, C. (2000). A note on the Utility of Incremental Learning. *AI Communications 13*(4), 215-223

Guajardo, J., Weber, R., & Miranda, J. (2006). A Forecasting Methodology Using Support Vector Regression and Dynamic Feature Selection. *Journal of Information & Knowledge Management 5*(4), 329–335.

Huang, W. (2007). *Temporal and Spatio-Temporal Data Mining*. IGI Publishing, Hershey.

Joentgen, A., Mikenina, L., Weber, R., & Zimmermann, H.-J. (1999). Dynamic Fuzzy Data Analysis Based on Similarity Between Functions. *Fuzzy Sets and Systems 105*(1), 81-90.

Kasabov, N. K., & Song, Q. (2002). DENFIS: Dynamic Evolving Neural-Fuzzy Inference System and Its Application for Time-Series Prediction. *IEEE Transactions on Fuzzy Systems 10*(2), 144-154.

Piatetsky-Shapiro, G. (1991). Knowledge Discovery in Real Databases: A Report on the IJCAI-89 Workshop. *AI Magazine 11*(5), 68-70.

Vanderveldt, I. V., Black, & Ch. L. (2001). *System and method for dynamic data-mining and on-line communication of customized information*. Patent Number 6266668, Issue date: July 24, 2001, Current US Classification: 707/10; 707/100; 706/15, International Classification: G06F 1730.

Weber, R. (2007). Fuzzy Clustering in Dynamic Data Mining: Techniques and Applications. In: J. Valente de Oliveira and W. Pedrycz (Editors): Advances in Fuzzy Clustering and Its Applications. John Wiley and Sons, 315-332.

Weber, R., & Guajardo, J. (2007). *Dynamic Data Mining for Improved Forecasting in Logistics and Supply Chain Management*. In: Haasis, H.-D., Kreowski, H.-J., Scholz-Reiter, B. (Eds.), Dynamics in Logistics, Proceedings of the First Int. Conf. LDIC 2007, Springer 2008, 55-61.

KEY TERMS

Dynamic Data Mining: Area within data mining that deals with dynamic aspects in any of the steps of the KDD process.

Evolving Stream Data Mining: Mining data streams that change their pattern unexpectedly.

Incremental Data Mining: Mining data that become available over time, usually one at a time.

Model Updating: Adaptation of an existing data mining model when undesired deviations during the monitoring of performed applications occur.

Stream Data Mining: We refer to stream data mining when it comes to mine in data that arrives in enormous quantities under memory constraints thus making it necessary to process data in only one direction.

Temporal Data Mining: Temporal data is classified to either categorical event streams or numerical time series.

Dynamical Feature Extraction from Brain Activity Time Series

Chang-Chia Liu
University of Florida, USA

W. Art Chaovalitwongse
Rutgers University, USA

Panos M. Pardalos
University of Florida, USA

Basim M. Uthman
NF/SG VHS & University of Florida, USA

INTRODUCTION

Neurologists typically study the brain activity through acquired biomarker signals such as Electroencephalograms (EEGs) which have been widely used to capture the interactions between neurons or groups of neurons. Detecting and identifying the abnormal patterns through visual inspection of EEG signals are extremely challenging and require constant attention for well trained and experienced specialists. To resolve this problem, data mining techniques have been successfully applied to the analysis for EEG recordings and other biomarker data sets. For example, Chaovalitwongse et al., (2006; 2007), Prokopyev et al., (2007) and Pardalos et al., (2007) reported the EEG patterns can be classified through dynamical features extracted from the underlying EEG dynamical characteristics. Moreover, in the area of cancer research, Busygin et al., (2006) showed promising results via Bi-clustering data classification technique using selected features from DNA microarrays. Ceci et al., (2007); Krishnamoorthy et al., (2007) also reported that data mining techniques enable protein structure characterization and protein structure prediction. From data mining aspects, feature extraction and selection for time series data sets not only play an important role in data preprocessing but also provide opportunities to uncover the underlying mechanisms of data under study. It also keeps the essential data structure and makes a better representation of acquired data sets that need to be classified.

In this work, the properties and descriptions of the most common neurological biomarker namely EEG data sets will be given as well as the motivations and challenges for abnormal EEG classification. The dynamical features for EEG classification will be reviewed and described in the second part of this work. The described dynamical features can also be applied to other types of classification applications for discovering the useful knowledge from obtained data sets. Finally, the potential applications for EEG classification will be discussed and comments for further research directions will be given in the future trends and conclusion sections.

BACKGROUND

Epilepsy is a common neurological disorder, affecting about 50 million people worldwide (WHO, 2005). Anti antiepileptic drugs (AEDs) are the mainstay of contemporary treatment for epilepsy, it can reduce frequency of seizure and prevent seizure recurrence in most cases. For subjects with uncontrolled seizures, ablative surgery is considered after two or more AEDs have failed to result for seizure freedom. The EEG (see Fig. 1 for an example) is a key tool in diagnosing seizure disorders and contributing tremendously to surgical decisions in patients with intractable epilepsy. The EEG recordings provide information about underlying interactions among neurons around the recording electrodes as a function of time. By carefully investigating EEG patterns, the spatio-temporal neuronal electrical activities can be decoded and abnormal patterns can be captured for diagnostic purposes (Berger, 1929). Through nonlinear dynamical features, data mining techniques have

Figure 1. An example of 10 second, 32-channel intracranial EEG display

made progresses in shedding light on hidden patterns in EEG recordings for such neurological disorders (Chaovalitwongse et al., 2006; 2007).

Nonlinear nature of EEG recordings has been shown by multiple researchers (Casdagli et al., 1997; Liu et al., 2007), features generated from nonlinear dynamics have also been shown to have high applicability for EEG analysis with promising results both on classifying and predicting epileptic seizures in EEG recordings (Iasemidis et al., 1990, 2003(a),2003(b), 2004; Le Van Quyen et al., 1998, 2005; Sackellares et al., 2006; Chaovalitwongse et al., 2006, 2007; Pardalos et al., 2006, 2007). The abnormal activities detected by nonlinear dynamical methods which are not able to achieve by traditional linear measurements. Classifying through these nonlinear characteristics in the feature space, the abnormal EEG pattern can be exploited and distinguished from the normal activity in the brain.

MAIN FOCUS

The spatio-temporal nonlinear dynamics has been steadily involved in EEG analysis from its conception in the 1970s, to proof-of-principle experiments in the late 1980s and 1990s, to its current place as an area of vigorous, clinical and laboratory investigation. A specific standard for future epilepsy research was to validate the presences of abnormal brain activities and eventually link them with treatment strategies that interrupt the abnormal activities (Seref et al., 2007). Results from many studies in data mining fluctuate broadly from their theoretical approaches to the problem, the amount of data analyzed and validation of the results. Relative weaknesses in this field include the lack of extensive testing on baseline data, the lack of technically rigorous validation and quantification of algorithm performance in many published studies, and the lack of methods for extracting the useful information from multi-channel and multi-feature data sets for the data per-processing. Classifications through constructed feature space, the domain of the system and solution to

that problem are given more insightful comprehension about the underlying processes under study. Using novel dynamical features from field of nonlinear dynamics, our group has successfully reported on predicting incoming seizures, detecting abnormal brain activities and capturing the effects of therapeutic interventions. Other researchers have also documented the usefulness of nonlinear dynamics for conducting brain research (Elger et al., 1998; Lehnertz et al., 2007; L.V. Quyen et al., 1998, 2005; Mormann et al., 2005, 2006).). In this section, three dynamical features will be described together with computational results obtained from real EEG data sets obtained from patients with epilepsy.

Lyapunov exponent: The maximum Lyapunov exponent is one of the most commonly used methods for estimating the stability of nonlinear chaotic systems like EEG recordings. Lyapunov exponents quantify the rate of divergence or convergence of two nearby initial points of a dynamical system, in a global sense. A positive Lyapunov exponent measures the average exponential divergence of two nearby trajectories, whereas a negative Lyapunov exponent measures exponential convergence of two nearby trajectories. If a system is chaotic, its largest Lyapunov exponent Lmax would be positive. The Lmax is defined as the average of local Lyapunov exponents L_{ij} in the state space, defined as follows:

$$L_{max} = \frac{1}{N_\alpha} \cdot \sum_{\alpha} L_{ij},$$

where N_α is the total number of the local Lyapunov exponents that are estimated from the evolution of adjacent points (vectors), $Y_{i=}Y(t_i)$ and $Y_{j=}Y(t_j)$ in the state space such that

$$L_{ij} = \frac{1}{\Delta t} \cdot \log_2 \frac{|Y(t_i + \Delta t) - Y(t_j + \Delta t)|}{|Y(t_i) - Y(t_j)|},$$

where Δt is the evolution time allowed for the vector difference $\delta_{xk}^{(\lambda)} = |X(t_i) - X(t_j)|$ to evolve to the new difference $\delta k\,(x_k) = |Y(t_i + \Delta t) - Y(t_j + \Delta t)|$, where $\lambda = 1...$ $N\alpha - 1$ and $\Delta t = k \cdot dt$ and dt is the sampling period of the original time series. If Δt is given in second, Lmax is in bits/sec.

Iasemidis et al. (1990) first introduced Lyapunov exponent for analyzing the EEG data sets recorded from patients with epilepsy. The Lyapunov exponent has been reported to have the ability to predict epileptic seizures about 1 hour prior to seizure onset (Iasemidis et al., 2003, 2004, 2005).

Minimum Embedding Dimension: Dynamical systems processing d degrees of freedom may choose to evolve on a manifold of much lower dimension, so that only fractions of the total number of degrees of

Figure 2. A plot demonstrates the values of the Lmax start decreasing 2 hours prior to seizure onset

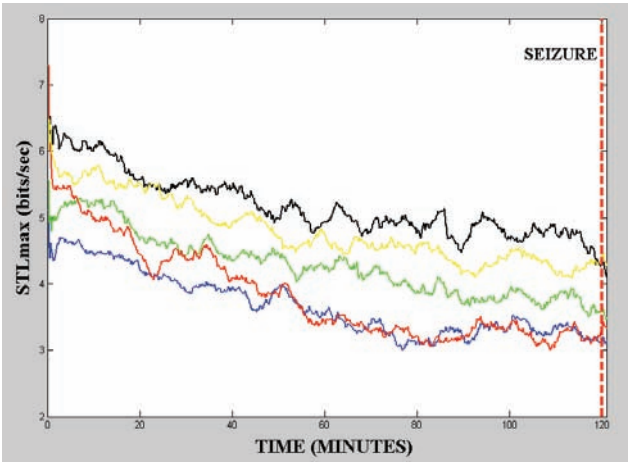
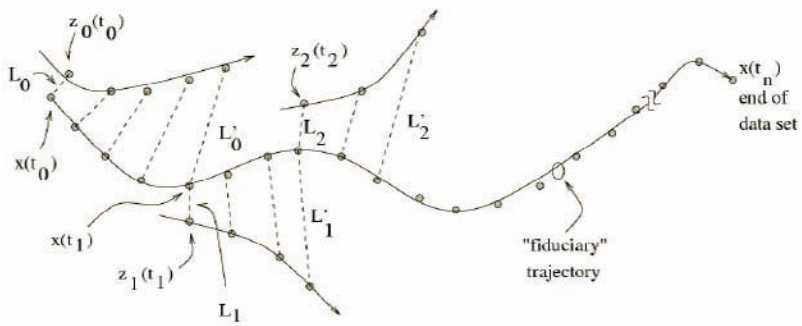


Figure 1. The steps for finding Lmax on an attractor



freedom are active. In such case it is useful to know how many degrees of freedom are actually active, and it is obvious that this information can be obtained from that dimension of attractor of the corresponding system. Choosing a too low value of the embedding dimension will result in points that are far apart in the original phase space being moved closer together in the reconstruction space. Takens' delay embedding theorem (Taken, 1980) states that a pseudo-state space can be reconstructed from infinite noiseless time series (when one chooses d>2dA) and can be used when reconstructing the delay vector. The classical approaches usually require huge computation power and vast amounts of data. We evaluated the minimum embedding dimension of the EEG recordings by using modified false neighbor method (Cao et al., 1997). Suppose that we have a time series $\{X_1, X_2 \ldots X_n\}$. By applying the method of delay we obtain the time delay vector, we can obtain the time-delay vector:

$$y_i(d) = (x_i, x_{i+\tau}, \ldots, x_{i+(d-1)\tau}) \text{ for i= } 1,2 \ldots N(d-1)\tau,$$

where d is the embedding dimension and τ is the time-delay and $y_i(d)$ means the i^{th} reconstructed vector with embedding dimension d. Similar to the idea of the false neighbor methods, defining

$$a(i,d) = \frac{\|y_i(d+1) - y_{n(i,d)}(d+1)\|}{\|y_i(d) - y_{n(i,d)}(d)\|} \text{ for } i = 1,2,\ldots,N - d\tau$$

where $\|\cdot\|$ is some measurement of Euclidian distance and given in this paper is the maximum norm. Define the mean value of all $a(i,d)$'s as

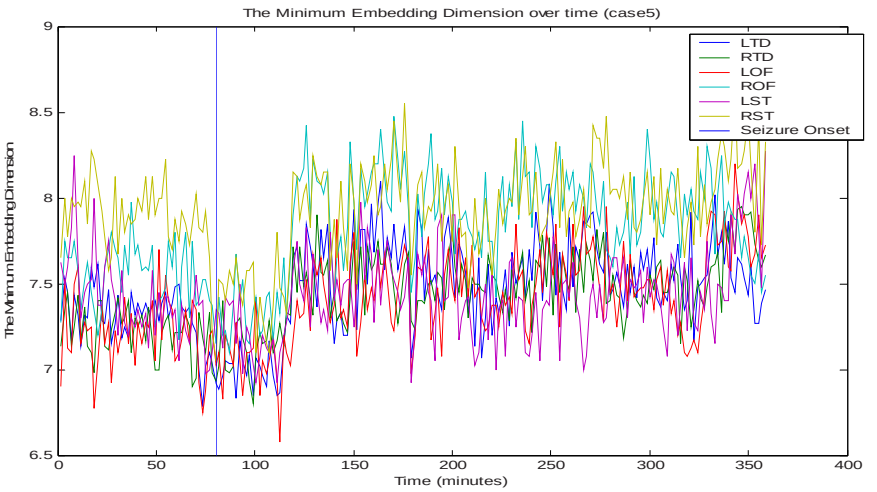
$$E(d) = \frac{1}{N-d\tau}\sum_{i=1}^{N-d\tau} a(i,d)$$

.

$E(d)$ will only depend on the dimension d and the time delay τ. The minimum embedding dimension is determined (when $E1(d) = \frac{E(d+1)}{E(d)}$ close to 1) when d is greater than some value d_0. If the time series comes from an attractor, then $d_0 + 1$ is the minimum embedding dimension. Liu et al. (2007) reported encouraging results for early seizure detection for the data sets acquired from patients with temporal lobe epilepsy.

Approximate Entropy: The Approximate entropy (ApEn) is a measure of complexity for systems. It was introduced by Pincus (1991) and has been widely applied in medical data. It can be used to differentiate between normal and abnormal data in instances where moment statistics (e.g. mean and variance) approaches fail to show a significant difference. Applications include heart rate analysis in the human neonate and in epileptic activity in electrocardiograms (Diambra, 1999). Mathematically, as part of a general theoretical framework, ApEn has been shown to be the rate of approximating a Markov chain process (Pincus, 1993). Most importantly, compared with Kolmogorov-Sinai (K-S) entropy (Kolmogorov, 1958), ApEn is generally finite and has been shown to classify the complexity of a system via fewer data points using theoretical analyses

Figure 3. The minimum embedding dimension generated from EEG data sets over 350 minutes, the vertical line denote an onset of an epileptic seizure



of both stochastic and deterministic chaotic processes and clinical applications (Pincus er al., , 1991; Kaplan et al., 1991; Rukhin et al., 2000).

The calculation of ApEn of a signal s of finite length N is performed as follows. First fix a positive integer m and a positive real number r_f. Next, from the signal s the N–m+1 vectors $x_m(i) = \{s(i), s(i + 1), ..., s(i + m – 1)\}$ are formed. For each i, $1 \le i \le (N – m + 1)$the quantity $C_i^m(r_f)$ is generated as:

$$C_i^m(r_f) \frac{Number\ of\ such\ j\ that\ d(\{(i),x_m(i),x_m(j)\}\le r_f}{N-m+1} ,$$

where the distance d between vectors $x_m(i)$ and $x_m(j)$ is defined as:

$$d[x_m(i),x_m(j)] = \max_{k=1,2,...,m}\{|(s(i+k-1)-s(j+k-1)|\}$$

Next the quantity $\Phi(r_f)$ is calculated as:

$$\Phi^m(r_f) = \frac{1}{N-m+1}\sum_{i=1}^{N-m+1}\log C_i^m(r_f)$$

Finally the ApEn is defined as:

$$ApEn(m,r_f,N) = \Phi^m(r_f) - \Phi^{m+1}(r_f)$$

The parameter r_f corresponds to an a priori fixed distance between neighboring trajectory points and frequently, r_f is chosen according to the signal's standard deviation (SD). Hence, r_f can be viewed as a filtering level and the parameter m is the embedding dimension determining the dimension of the phase space.

An example figure shows *ApEn* generated from EEG data sets over 100 minutes, the vertical dash lines denote an epileptic seizure.

FUTURE TRENDS

Electroencephalographic recordings are among the most accessible neurological continuous time series data. They contain many complex and hidden patterns that might link to different neurological disorders. The main message of this chapter is the importance of extracting useful dynamical features in the analysis of EEG classification. Dynamical feature extraction techniques have been shown capable of characterizing EEG patterns at the microscopic level (Pardalos et al., 2002). With advanced high frequency EEG acquisition devices, the sampling rate can be easily set to as high as 10 KHz allowing us to capture 10,000 data points of neuronal activity per second. For longer EEG data sets, the amount of data and the computational time for analysis will not only increase dramatically but also make it almost impossible to analyze the entire data sets. Extraction of dynamical features offers a lower dimensional feature space, in contrast to work on raw EEG data sets directly. The use of dynamical features may allow the data mining techniques to operate efficiently in a nonlinear transformed feature space without being adversely affected by dimensionality of the feature space. Moreover, through specific and suitable computational data mining analysis the dynamical features will allow us to link mining results to EEG patterns with neurological disorders. In the future, it is important to bridge the classification results with clinical outcomes as this has been one of the most important tasks for conceiving automatic diagnoses schemes and controlling the progression of diseases through data mining techniques.

CONCLUSION

In this chapter, we used several dynamical features on the acquired EEG data sets to identify the abnormal patterns that were most important for discriminating between normal and abnormal brain activities. Moreover, results from our classification showed the dynamical features are also resulted in more meaningful outcomes for interpreting brain disorders.

REFERENCES

Berger, H. (1929). Über das Elektroenkephalogramm des Menschen Archiv für Psychiatrie und Nervenkrankheiten, 87.

Busygin, S. and Pardalos, P.M. (2007). Exploring microarray data with correspondence analysis. In P.M. Pardalos, V.L. Boginski, A. Vazacopolos (Eds.), Data mining in biomedicine (25-37). New Youk: Springer.

Cao, L. (1997). Practical method for determining the minimum embedding dimension of a scalar time series. *Physcai D*, 110, 43-50.

Casdagli, M., Iasemidis L. D., Savit, R. S., Gilmore, R. L., Roper, S. N., Sackellares, J. C. (1997). Non-linearity in invasive EEG recordings from patients with temporal lobe epilepsy. EEG Clin. Neurophysiol. 102 98-105.

Ceci, G, Mucherino, A., D'Apuzzo, M., Serafino, D.D., Costantini, S., Facchiano, A., Colonna, G., (2007). Computational Methods for Portein fold prediction: an Ab-initio topological approach. In P.M. Pardalos, V.L. Boginski, A. Vazacopolos (Eds.), Data mining in biomedicine (391-429). New Youk:Springer.

Chaovalitwongse, W., Iasemidis, L. D., Pardalos, P. M., Carney, P. R., Shiau, D. S., & Sackellares, J. C. (2005). Performance of a seizure warning algorithm based on the dynamics of intracranial EEG. Epilepsy Res, 64(3), 93-113.

Chaovalitwongse, W. A., Iasemidis, L. D., Pardalos, P. M., Carney, P. R., Shiau, D. S., & Sackellares, J. C. (2006). Reply to comments on "Performance of a seizure warning algorithm based on the dynamics of intracranial EEG" by Mormann, F., Elger, C.E., and Lehnertz, K. Epilepsy Res, 72(1), 85-87.

Diambra, L., Bastos de Figueiredo , J.C., Malta, C.P. (1999). Epileptic activity recognition in EEG recording. Physica A, 273, 495-505.

Elger, C. E., & Lehnertz, K. (1998). Seizure prediction by non-linear time series analysis of brain electrical activity. Eur J Neurosci, 10(2), 786-789.

Krishnamoorthy, B. Provan, S., Tropsha, A. (2007). A topolidical characterization of protein structure. In P.M. Pardalos, V.L. Boginski, A. Vazacopolos (Eds.), Data mining in biomedicine (431-455). New Youk: Springer.

Iasemidis, L. D., Sackellares, J. C., Zaveri, H. P., & Williams, W. J. (1990). Phase space topography and the Lyapunov exponent of electrocorticograms in partial seizures. Brain Topogr, 2(3), 187-201.

Iasemidis, L. D., Shiau, D. S., Chaovalitwongse, W., Sackellares, J. C., Pardalos, P. M., Principe, J. C., et al. (2003). Adaptive epileptic seizure prediction system. IEEE Trans Biomed Eng, 50(5), 616-627.

Iasemidis, L. D. (2003). Epileptic seizure prediction and control. IEEE Trans Biomed Eng, 50(5), 549-558.

Iasemidis, L. D., Shiau, D. S., Sackellares, J. C., Pardalos, P. M., & Prasad, A. (2004). Dynamical resetting of the human brain at epileptic seizures: application of nonlinear dynamics and global optimization techniques. IEEE Trans Biomed Eng, 51(3), 493-506.

Iasemidis, L. D., Shiau, D. S., Pardalos, P. M., Chaovalitwongse, W., Narayanan, K., Prasad, A., et al. (2005). Long-term prospective on-line real-time seizure prediction. Clin Neurophysiol, 116(3), 532-544.

Kaplan, D.T., Furman, M.I., Pincus, S.M., Ryan, S.M., Lipsitz , L.A., Goldberger, A.L., (1991). Aging and the complexity of cardiovascular dynamics. *Biophys. J.*, 59, 945–949.

Kolmogorov , A.N., (1958). A new metric invariant of transitive dynamical systems, and Lebesgue space automorphisms. *Dokl. Akad. Nauk SSSR* , l19, 861–864.

Le Van Quyen, M., Adam, C., Baulac, M., Martinerie, J., & Varela, F. J. (1998). Nonlinear interdependencies of EEG signals in human intracranially recorded temporal lobe seizures. Brain Res, 792(1), 24-40.

Le Van Quyen,M., Soss, J., Navarro V., Robertson, R., Chavez, M., Baulac, M., Martinerie, J. (2005). Preictal state identication by synchronization changes in lonterm intracranial EEG recordings. J Clin Neurophysiology, 116, 559-568.

Lehnertz, K., Mormann, F., Osterhage, H., Muller, A., Prusseit, J., Chernihovskyi, A., et al. (2007). State-of-the-art of seizure prediction. J Clin Neurophysiol, 24(2), 147-153.

Liu, C.-C., Pardalos, P.M., Chaovalitwongse, W., Shiau, D. S., Yatsenko, V.A., Ghacibeh, G., Sackellares, J.C. (2007). Quantitative complexity analysis in multi-channel intercranial EEG recordings from epilepsy brains. J. Combinatorial Optimization, to be appeared.

Liu, C.-C., Shiau, D. S., Pardalos, P.M., Chaovalitwongse, W., Sackellares, J.C. (2007). Presence of nonlinearity in intracranial EEG recordings: detected by Lyapunov exponents. AIP conference proceedings, 953, 197-205

Mormann, F., Elger, C. E., & Lehnertz, K. (2006). Seizure anticipation: from algorithms to clinical practice. Curr Opin Neurol, 19(2), 187-193.

Mormann, F., Kreuz, T., Rieke, C., Andrzejak, R. G., Kraskov, A., David, P., et al. (2005). On the predictability of epileptic seizures. Clin Neurophysiol, 116(3), 569-587.

Seref, O., Kundakciogli, O. E., Pardalos, P.M. (Eds.). (2007). Data mining, systems analysis, and optimization in biomedicine. AIP conference proceedings vol. 953. New York: American Institute of Physics.

Pardalos, P.M., Jose Principe (Eds.). (2002). Biocomputing. Massachusetts: Kluwer Academic Publishers,

Pardalos, P.M., Boginski, V.L., Vazacopulos, A. (Eds.). (2007). Data mining in biomedicine. New York: Springer.

Pincus, S.M. (1991) Approximate entropy as a measure of system complexity. *Proc Natl Acad Sci U S A*, 88, 2279-2301

Pincus, S.M., Gladstone, I.M., Ehrenkranz, R.A., (1991). A regularity statistic for medical data analysis. *J Clin Monit.*, 7, 335–345.

Pincus S.M., Cummins T.R., Haddad , G.G., (1993). Heart rate control in normal and aborted SIDS infants. *Am. J. Physiol.*, vol. 264,638-646.

Prokopyev, O.A., Busygin, S., Pardalos, P.M., (2007). An Optimization Based Approach for Data Classification. *Optimization Methods and Software*, Vol. 22/1, pp. 3–9.

Rukhin, A. L. (2000). Approxmate evtropy for testing randomness. J. Appl. Proba, 37, 88-100.

Sackellares, J. C., Shiau, D. S., Principe, J. C., Yang, M. C., Dance, L. K., Suharitdamrong, W., et al. (2006). Predictability analysis for an automated seizure prediction algorithm. J Clin Neurophysiol, 23(6), 509-520.

Takens, F. (1980). Drtecting strange attractors in turbulence. In Lecture Notes in Math.,D. A. Rand & L.S. Young, Eds., Vol 898, pp336-381, Springer-Verlag

World Health Organization, International League Against Epilepsy, International Bereau for Epilepsy (2005). Atlas: Epilepsy care in the world. WHO library cataloguing-in publication data.

KEY TERMS

Attractor: An attractor is defined as a phase space point or a set of points representing the various possible steady-state conditions of a system; an equilibrium state or a group of states to which a dynamical system converges and cannot be decomposed into two or more attractors with distinct basins of attraction.

Degree of Freedom: In the context of dynamical systems, the degree of freedom is the number of variables needed to describe the underlying system.

Fixed Point: A point to which different trajectories in phase space tend to at rest is called a fixed point.

Limit Cycle: A limit-cycle is a closed trajectory in phase space having the property that at least one other trajectory spirals into it either as time approaches infinity.

Phase Space: Phase space is the collection of possible states of a dynamical system. In general, the phase space is identified with a topological manifold. An n-dimensional phase space is spanned by a set of n-dimensional "embedding vectors", each one defining a point in the phase space, thus representing the instantaneous state of the system.

Strange Attractor: A strange attractor is defined as an attractor that shows sensitivity to initial conditions (exponential divergence of neighboring trajectories) and that, therefore, occurs only in the chaotic domain. While all chaotic attractors are strange, not all strange attractors are chaotic. In other words, the chaoticity condition is necessary, but not sufficient, for the strangeness condition of attractors.

Trajectory: Trajectory or orbit of the dynamical system is the path connecting points in chronological order in phase space traced out by a solution of an initial value problem. If the state variables take real values in a continuum, the orbit of a continuous time system is a curve, while the orbit of a discrete-time system is a sequence of points.

Efficient Graph Matching

Diego Reforgiato Recupero
University of Catania, Italy

INTRODUCTION

Application domains such as bioinformatics and web technology represent complex objects as graphs where nodes represent basic objects (i.e. atoms, web pages etc.) and edges model relations among them. In biochemical databases proteins are naturally represented as labeled graphs: the nodes are atoms and the edges are chemical links. In computer vision, graphs can be used to represent images at different levels of abstraction. In a low-level representation (Pailloncy, Deruyver, & Jolion, 1999), the nodes of a graph correspond to pixels and the edges to spatial relations between pixels. At higher levels of description (Hong & Huang, 2001), nodes are image regions and edges are spatial relations among them. In a Web graph (Deutsch, Fernandez, Florescu, Levy, & Suciu, 1999) nodes are web pages and edges are links between them. In all these domains, substructure queries that search for all exact or approximate occurrences of a given query graph in the graphs of the database can be useful.

Research efforts in graph searching have taken three directions: the first is to study matching algorithms for particular graph structures (planar graphs, bounded valence graphs and association graphs); the second is to use elaborate tricks to reduce the number of generated matching maps (Cordella, Foggia, Sansone, & Vento, 2004); and the third is, since graph searching problem is NP-complete, to provide polynomial approximate algorithms. In the context of querying in a database of graphs many of the existing methods are designed for specific applications. For example, several querying methods for semi-structured databases have been proposed. In addition, commercial products and academic projects (Kelley 2002) for subgraph searching in biochemical databases are available. These two examples have a different underlying data model (a web-page database is a large graph whereas a biochemical database is a collection of graphs). In these two applications regular path expressions and indexing methods are used during query time to respectively locate substructures in the database and to avoid unnecessary traversals of the database. In general graph databases, there are some searching methods where the data graph and the query graph must have the same size. Other methods allow the query to be considerably smaller than the database graphs. A common idea in the above algorithms is to index the subgraph matches in the database and organize them in suitable data structures.

BACKGROUND

A graph database can be viewed as either a single (large) labeled graph (e.g. web) or a collection of labeled graphs (e.g., chemical molecules). By keygraph searching it is intended graph or subgraph matching in a graph database. Although (sub)graph-to-graph matching algorithms can be used for keygraph searching, efficiency considerations suggest the use of indexing techniques to reduce the search space and the time complexity especially in large databases. Keygraph searching in databases consists of three basic steps:

1. Reduce the search space by filtering. For a database of graphs a filter finds the most relevant graphs; for a single-graph database it identifies the most relevant subgraphs. In this work, the attention is restricted to filtering techniques based on the structure of the labeled graphs (paths, subgraphs). Since searching for subgraph structures is quite difficult, most algorithms choose to locate paths of node labels.
2. Formulate query into simple structures. The query graph can be given directly as a set of nodes and edges or as the intersection of a set of paths. Furthermore the query can contain wildcards (representing nodes or paths) to allow more general searches. This step normally reduces the query graph to a collection of small paths.

3. Match. Matching is implemented by either traditional (sub)graph-to-graph matching techniques, or combining the set of paths resulting from processing the path expressions in the query through the database.

A well-known graph-based data mining algorithm able to mine structural information is Subdue (Ketkar, Holder, Cook, Shah, & Coble, 2005). Subdue has been applied to discovery and search for subgraphs in protein databases, image databases, Chinese character databases, CAD circuit data and software source code. Furthermore, an extension of Subdue, SWSE (Rakhshan, Holder & Cook, 2004), has been applied to hypertext data. Yan et al. (Yan, Yu & Han, 2004) proposed a novel indexing technique for graph databases based on reducing the space dimension with the most representative substructures chosen by mining techniques. However, Subdue based systems accept one labeled graph at a time as input.

GRACE, (Srinivasa, Maier & Mutalikdesai, 2005) is a system where a data manipulation language is proposed for graph database and it is integrated with structural indexes in the DB.

Daylight (James, Weininger & Delany, 2000) is a system used to retrieve substructures in databases of molecules where each molecule is represented by a graph. Daylight uses fingerprinting to find relevant graphs from a database. Each graph is associated with a fixed-size bit vector, called the fingerprint of the graph. The similarity of two graphs is computed by comparing their fingerprints. The search space is filtered by comparing the fingerprint of the query graph with the fingerprint of each graph in the database. Queries can include wildcards. For most queries, the matching is implemented using application-specific techniques. However queries including wildcards may require exhaustive graph traversals. A free and efficient academic emulation of Daylight, called Frowns (Kelley, 2002), uses the compacted hashed paths vector filtering of Daylight and the subgraph matching algorithm VF (Cordella, Foggia, Sansone, & Vento, 2004).

In character recognition, a practical optical character reader is required to deal with both fonts and complex designed fonts; authors in (Omachi, Megawa, & Aso, 2007) proposed a graph matching technique to recognize decorative characters by structural analysis. In image recognition, an improved exact matching method using a genetic algorithm is described in (Auwatanamongkol, 2007).

One way to use a simple theoretical enumeration algorithm to find the occurrences of a query graph Ga in a data graph Gb is to generate all possible maps between the nodes of the two graphs and to check whether each generated map is a match. All the maps can be represented using a state-space representation tree: a node represents a pair of matched nodes; a path from the root down to a leaf represents a map between the two graphs. In (Ullmann, 1976) the author presented an algorithm for an exact subgraph matching based on state space searching with backtracking. Based on the Ullmann's algorithm, Cordella et al. (Cordella, Foggia, Sansone, & Vento, 2004) proposed an efficient algorithm for graph and subgraph matching using more selective feasibility rules to cut the state search space. Foggia et al. (Foggia, Sansone, & Vento, 2001) reported a performance comparison of the above algorithm with other different algorithms in literature. They showed that, so far, no one algorithm is more efficient than all the others on all kinds of input graphs.

MAIN FOCUS

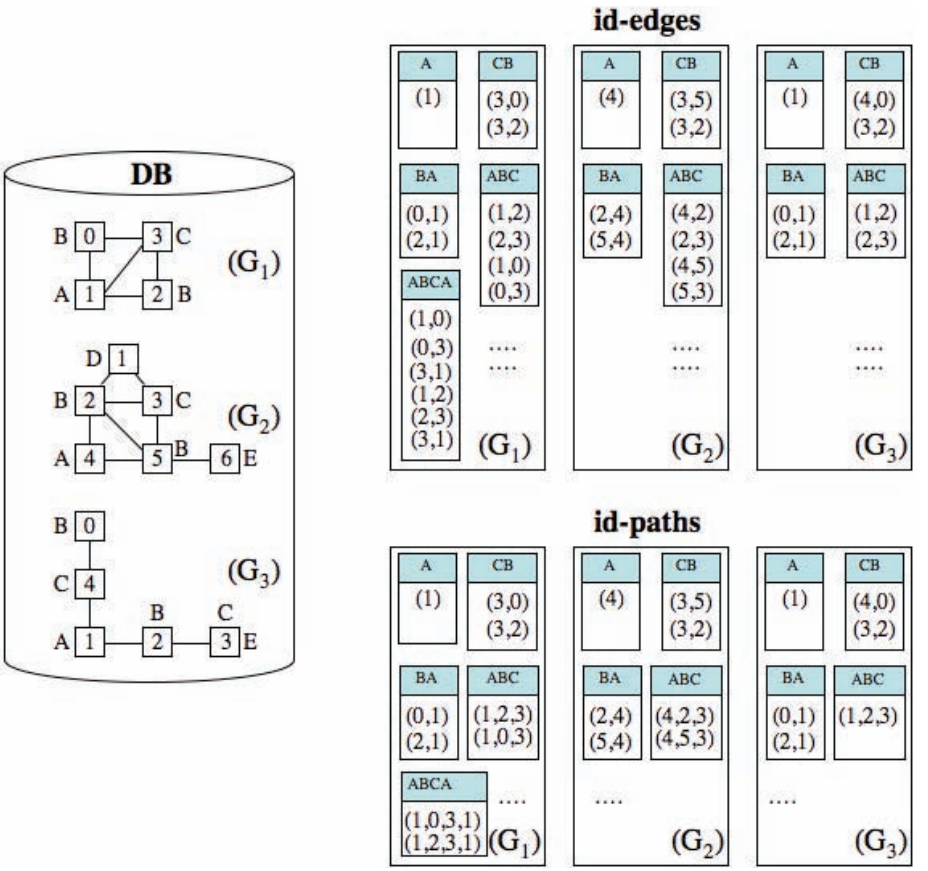
The idea of an efficient method for graph matching is to first represent the label paths and the label edges of each graph of the database as compacted hash functions; then, given the query graph, the paths of the database graphs not matching the query paths will be pruned out in order to speed up the searching process.

The proposed method models the nodes of data graphs as having an identification number (node-id) and a label (node-label). An id-path of length n is a list of $n+1$ node-ids with an edge between any two consecutive nodes. A label-path of length n is a list of $n+1$ node-labels. An id-path of length 1 is called id-edge. The label-paths and the id-paths of the graphs in a database are used to construct the index of the database and to store the data graphs. Figure 1 shows three input graphs with id-paths and id-edges.

Indexing Construction

Let l_p be a fixed positive integer (in practice, 4 is often used). For each graph in the database and for each node, all paths that start at this node and have length from one up to l_p are found. The index is implemented using a hash table. The keys of the hash table are the hash values of the label-paths. Collisions are solved

Figure 1. Three input graphs in Berkeley DB with their id-edges and id-paths for l_p=3.



by chaining, that is, each hash table bucket contains the linked list of the label-paths together with the number of their occurrences in each database graph. This hash table will be referred as the fingerprint of the database.

Path Representation of a Graph

Since several paths may contain the same label sequence, the id-edges of all the paths representing a label sequence are grouped into a label-path-set. The collection of such label-path-sets is called the path-representation of a graph.

Data Storing

The Berkeley DB (Berkeley) is used as the underlying database to store the massive data resulting from data graph representation and indexing construction. The fingerprint is stored as a dynamic Berkeley DB hash table of linked lists whose keys are the hashed label-path values. This allows efficient execution of the subgraph queries. Each graph is stored in a set of Berkeley DB tables each corresponding to a path and containing the set of id-edges comprising the occurrences of that path (label-path-set).

Queries Decomposition

The execution of a query is performed by first filtering out those graphs in the database which cannot contain the query. To do this, the query is parsed to build its fingerprint (hashed set of paths). Moreover, the query is decomposed into a set of intersecting paths in the following way: the branches of a depth-first traversal tree of the graph query are decomposed into sequences of overlapping label paths, called patterns, of length less than or equal to l_p. Overlaps may occur in the following cases:

- For consecutive label-paths, the last node of a pattern coincides with the first node of the next pattern (e.g. A/B/C/B/, with l_p=2, is decomposed into two patterns: A/B/C/ and C/B/);
- If a node has branches it is included in the first pattern of every branch;
- The first node visited in a cycle appears twice: in the beginning of the first pattern of the cycle and at the end of the last pattern of the cycle (the first and last pattern can be identical.) (See Figure 2.)

GrepVS Database Filtering

A first filtering is performed by comparing the fingerprint of the query with the fingerprint of the database. A graph, for which at least one value in its fingerprint is less than the corresponding value in the fingerprint of the query, is filtered out. The remaining graphs are candidates for matching. See Figure 3.

Then, parts of the candidate graphs are filtered as follows:

- decomposing the query into patterns;
- selecting only those Berkeley DB tables associated with patterns in the query.

The restored edge-ids correspond to one or several subgraphs of candidate graphs. Those subgraphs are the only ones that are matched with the query. Notice that the time complexity of filtering is linear in the size of the database. Figure 4 shows the subgraph filtering for the obtained graph in Figure 3.

The above method requires finding all the label-paths up to a length l_p starting from each node in the query graph. Depending on the size of the query the expected online querying time may result in a rather expensive task. In order to avoid that cost, one may choose to modify the fingerprint construction procedure as follows: only label-paths (patterns) found in the

Figure 2. A query graph (a); its depth first tree (b); patterns obtained with l_p=3. Overlapping labels are marked with asterisks and underlines. Labels with same mark represent the same node.

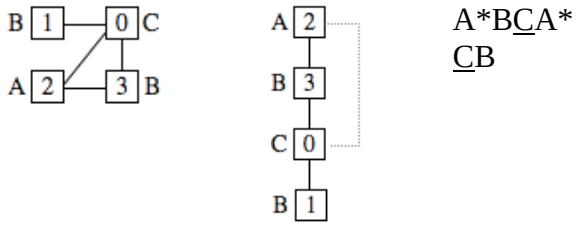


Figure 3. Filtering for graphs of Figure 1 and query of Figure 2

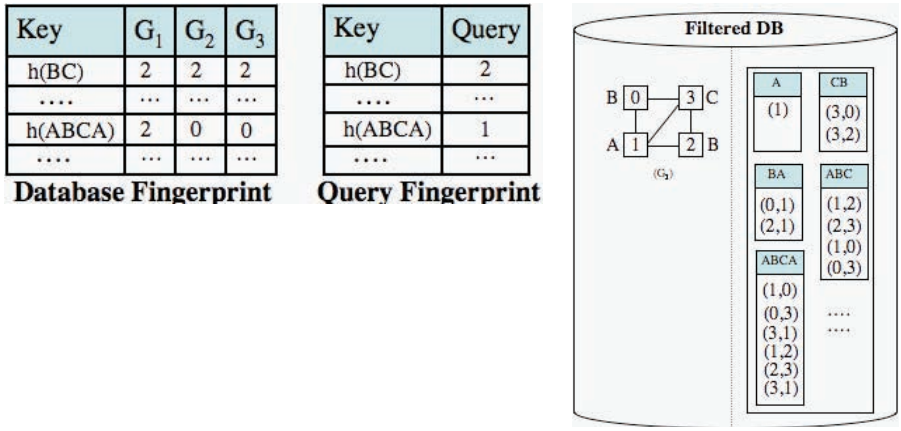
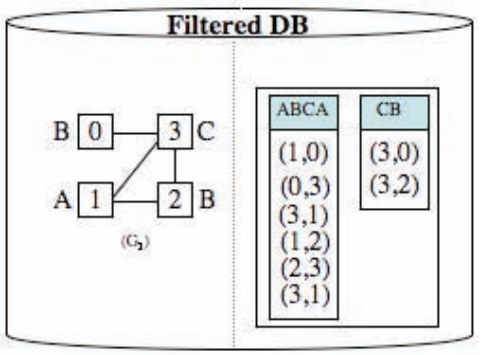


Figure 4. Subgraph filtering for graph in Figure 3.



decomposition step are used. In the proposed method, a single fingerprint value is associated to each pattern. Obviously, this method requires less time and space to filter, but is less selective. However, it may be required in some applications. The performance test considers this modified method.

GrepVS Subgraph Matching

After filtering, a collection of subgraph candidates is left (see Figure 4) which may contain matches with the query. Subgraph matching is performed by the VF (Cordella, Foggia, Sansone, & Vento. 2004) algorithm. VF algorithm is a refinement of Ullmann's subgraph isomorphism algorithm with a more selective feasibility rules to prune the state search space. Given a query graph *Q* and data graph *G*, the matching process is carried out by using the state space representation where a state is partial mapping, whereas a transition between two states corresponds to the addition of a new pair of matched nodes. The aim of the matching process is the determination of a mapping which is a bijection and consists of a set of node pairs covering all the query graph nodes. When a node pair is added to the partial mapping, coherence conditions are checked. If the new pair produces an incoherent state, then any further exploration of that path is avoided. The introduction of feasibility rules allows the pruning of the search space, reducing significantly the computational cost of the matching process. The feasibility rules are divided into semantic rules which guarantee the semantic coherence (they are satisfied if the labels of nodes corresponding in the mapping are equal) and three syntactic rules which guarantee the syntactic coherence. They are described in (Cordella, Foggia, Sansone, & Vento. 2004), where it is also stated that the computational complexity in the worst case of the VF algorithm is $\Theta(N!N)$ whereas its spatial complexity is $\Theta(N)$ where $N = |V1|+|V2|$.

Performance Analysis

GrepVS is implemented in ANSI C++ and it uses Berkeley DB as the underlying database.

To evaluate GrepVS both query and preprocessing time have been measured varying database size, number of nodes, edges and node labels on the input dataset, degree of the nodes, efficiency of filtering, and the parameter l_p.

Three kinds of databases have been considered:

1. Synthetic databases of size 1000 to 5000 graphs generated by (Kuramochi & Karypis. 2001). They are generated by a set of potentially frequent connected subgraphs; the number of distinct vertex labels is specified by a parameter; for each frequent connected subgraph, its topology as well as its vertex labels are randomly generated. (These will be referred to as random graphs).
2. Synthetic 2D mesh databases of size 1000 to 5000 graphs from (Foggia, Sansone, & Vento. 2001). (These will be referred to as mesh graphs).
3. Databases containing 10000 up to 50000 molecules available at (National Cancer Institute). (These will be referred to as AIDS graphs).

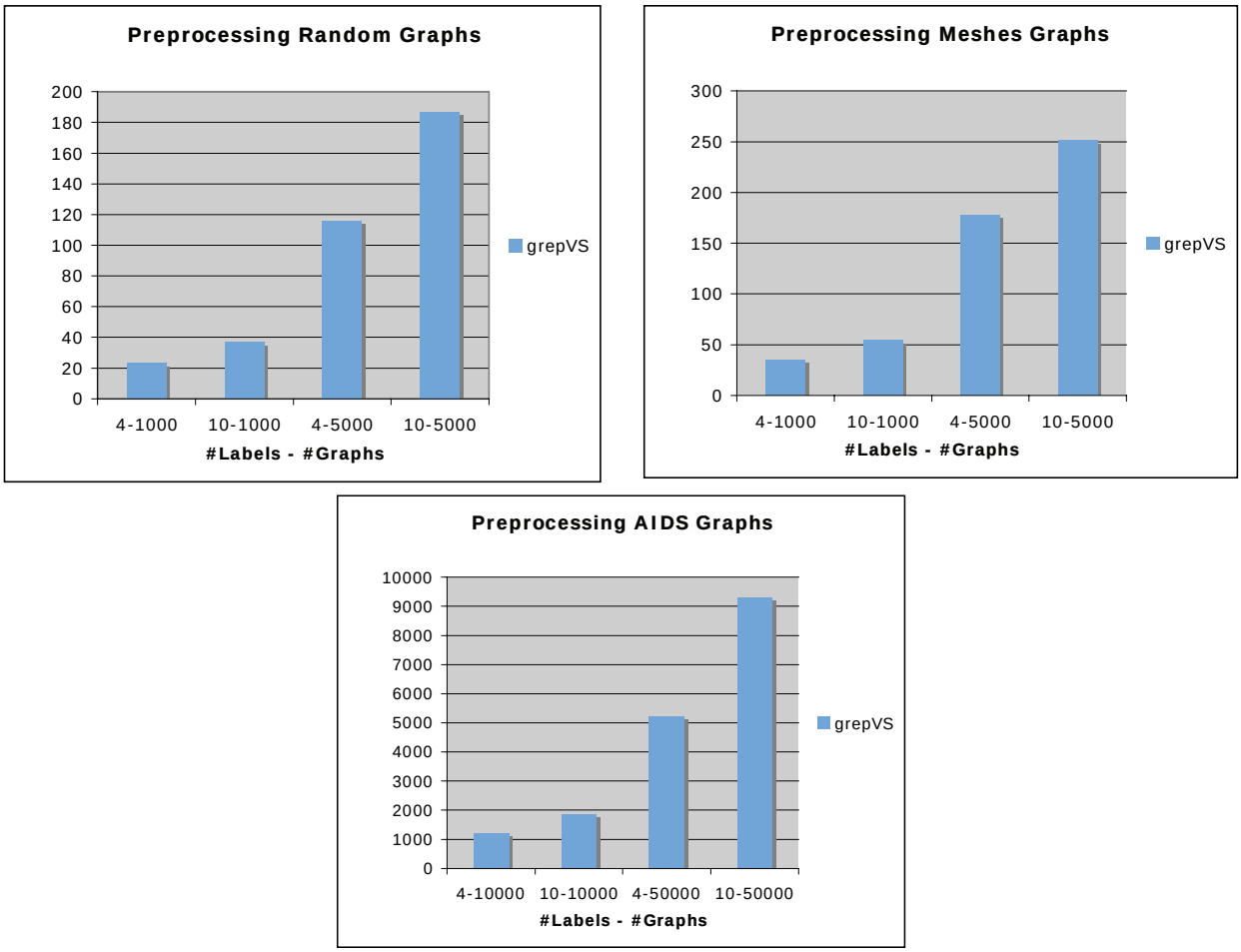
Building the database requires finding all the id-paths of length up to l_p and it is independent of the number of different labels in the graphs. Figure 5 shows the preprocessing time for each type of dataset considered.

On the other hand, the query running time is strongly influenced by the number of different labels. Increasing the number of labels in the graphs implies that fingerprints are more selective and the number of matches decreases.

Results

GrepVS has been compared against Subdue, Frowns, Daylight, and Grace for both single-graph databases and multi-graphs databases.

Figure 5. GrepVS preprocessing for the types of graphs considered.



GrepVS performs in a matter of seconds query searches that take Subdue more than 10 minutes. Moreover, GrepVS performs significantly better than Frowns, Daylight, and Grace. In each experiment four graphs with different numbers of nodes and edges have been generated. For meshes and random graph datasets each graph had a number of nodes ranging from 100 to 200, whereas the number of different labels ranged from 4 to 10. For AIDS graph datasets each graph had a number of nodes ranging from 100 to 5000, whereas the number of different labels was over 50. For AIDS databases the running times are faster due to the number of labels. On the latter the filtering process is much more effective.

FUTURE TRENDS

Graph matching techniques will become key components of any data mining instrument.

There are still a number of open issues that need further research. There are many applications in computer vision and pattern recognition where the full representational power of graphs may not be required. Focusing on special subclasses of graphs may result in more efficient matching procedures. Moreover, the design of efficient matching algorithms for very large graph queries will be of primary interest for graph matching of big graphs. For example, aligning a large protein interaction network against a set of other protein interaction networks.

CONCLUSION

GrepVS is an application-independent method for finding all the occurrences of a query graph within a database of graphs. This is an important problem in many applications from computer vision to chemistry. Several searching systems exist for chemical databases but not much research has been done for application-independent searching in database of graphs. GrepVS incorporates efficient searching algorithms and selective filtering techniques and allows inexpensive data analysis across many domains. It has been extensively tested on both synthetic databases and molecular databases. It appears to outperform the best-known alternative methods, from which it borrows many ideas and sometimes code.

REFERENCES

Auwatanamongkol, S. (2007). Inexact graph matching using a genetic algorithm for image recognition, *Pattern Recognition Letters, 28, 12, 1428-1437.*

Berkeley db, from http://www.sleepycat.com/

Cordella, L. P., Foggia, P., Sansone, C., & Vento, M. (2004). A (sub)graph isomorphism algorithm for matching large graphs, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10), 1367-1372.

Deutsch, A., Fernandez, M. F., Florescu, D., Levy, A. Y., & Suciu, D. (1999). A query language for xml, *Journal of Computer Networks* , 31(11-16), 1155-1169.

Foggia, P., Sansone, C., & Vento, M. (2001). A performance comparison of five algorithms for graph isomorphism. *Proceedings of the 3rd IAPR TC-15 Workshop on Graph-based Representations in Pattern Recognition*, 188-199.

Foggia, P., Sansone, C., & Vento, M. (2001). A database of graphs for isomorphism and sub-graph isomorphism benchmarking. *Proceedings of the 3rd IAPR TC-15 Workshop on Graph-based Representations in Pattern Recognition*, 176-187.

Hong P., & Huang, T. S. (2001). Spatial pattern discovering by learning the isomorphic subgraph from multiple attributed relational graphs. *Proceedings of ENTCS.*

James, C. A., Weininger, D., & Delany, J. (2000). Daylight theory manual-Daylight 4.71, www.daylight.com.

Kelley, B. (2002). Frowns, from http://staffa.wi.mit.edu/people/kelley/.

Kuramochi, M., & Karypis, G., (2001). Frequent subgraph discovery. *Proceedings of International Conference Data Mining*, 313–320.

Ketkar, N., Holder, L., Cook, D., Shah, R., & Coble, J. (2005). Subdue: Compression-based Frequent Pattern Discovery in Graph Data. *Proceedings of the ACM KDD Workshop on Open-Source Data Mining.*

National cancer institute, from http://www.nci.nih.gov.

Omachi, S., Megawa, S., & Aso, H. (2007). Decorative Character Recognition by Graph Matching. *IEICE Trans. on Information and Systems, E90, 10, 1720-1723.*

Pailloncy, J. G., & Deruyver A., & Jolion J. M. (1999). From pixels to predicates revisited in the graphs framework. *Proceedings of 2rd Int. Workshop on Graph based Represenatiation, GbR99.*

Rakhshan, A., Holder, L. B., & Cook, D. J. (2004). Structural Web Search Engine. *International Journal of Artificial Intelligence Tools.*

Srinivasa, S., Maier, M. & Mutalikdesai, M. R. (2005). LWI and Safari: A New Index Structure and Query Model for Graph Databases. *International Conference on Management of Data (COMAD 2005)*, 6-8.

Ullmann, J. R. (1976). An algorithm for subgraph isomorphism. *Journal of the Association for Computing Machinery*, 23, 31-42.

Yan, X., Yu, P. S., & Han, J. (2004). Graph indexing: A frequent structure based approach, *Proceedings of the ACM SIGMOD International Conference on Management of Data.*

Wang, S., & Wang, H. (2002). Knowledge discovery through self-organizing maps: Data visualization and query processing, *Knowledge and Information Systems*, 4(1), 31-45.

KEY TERMS

Approximate Graph Search: Given a query graph G with wildcards it is the problem of looking in each graph of a given dataset for possible sub-graphs matching G and satisfying the wildcards.

Exact Graph Search: Given a query graph G it is the problem of looking in each graph of a given dataset for possible sub-graphs matching G.

Filtering: A step consisting of removing irrelevant graphs from a given dataset.

Fingerprint: A data structure (usually a compacted hash-table) representing a database of graphs.

Graph Matching Problem: Given two graphs $G_a(V_a,E_a)$ and $G_b(V_b,E_b)$ the problem is to find a one-to-one mapping $f:V_a \rightarrow V_b$ such that $(u,v) \varepsilon E_s$ iff $(f(u),f(v)) \varepsilon E_b$.

Id-Path: A list of n+1 node-ids with an edge between any consecutive nodes.

Keygraph Searching: A graph or subgraph matching in a graph database.

Label-Path: A list of n+1 node labels.

Meshes Graphs: Graphs in which each node is connected to its 4 neighborhood nodes.

Semantic Coherence: The mapping of the labels of corresponding nodes of two graphs.

Syntactic Coherence: The mapping of the nodes and edges of two graphs.

Enclosing Machine Learning

Xunkai Wei
University of Air Force Engineering, China

Yinghong Li
University of Air Force Engineering, China

Yufei Li
University of Air Force Engineering, China

INTRODUCTION

As known to us, the cognition process is the instinct learning ability of the human being. This process is perhaps one of the most complex human behaviors. It is a highly efficient and intelligent information processing process. For a cognition process of the natural world, humans always transfer the feature information to the brain through their perception, and then the brain processes the feature information and remembers it. Due to the invention of computers, scientists are now working toward improving its artificial intelligence, and they hope that one day the computer could have its intelligent "brain" as human does. However, it is still a long way for us to go in order to let a computer truly "think" by itself.

Currently, artificial intelligence is an important and active research topic. It imitates the human brain using the idea of function equivalence. Traditionally, the neural computing and neural networks families are the majority parts of the direction (Haykin, 1994). By imitating the working mechanism of the human-brain neuron, scientists have built the neural networks theory following experimental research such as perception neurons and spiking neurons (Gerstner & Kistler, 2002) in order to understand the working mechanism of neurons.

Neural-computing and neural networks (NN) families (Bishop, 1995) have made great achievements in various aspects. Recently, statistical learning and support vector machines (SVM) (Vapnik, 1995) have drawn extensive attention and shown better performances in various areas (Li, Wei & Liu, 2004) than NN, which implies that artificial intelligence can also be made via advanced statistical computing theory. Nowadays, these two methods tend to merge under the statistical learning theory framework.

BACKGROUND

It should be noted that, for NN and SVM, the function imitation happens at microscopic view. Both of them utilize the mathematic model of neuron working mechanism. However, the whole cognition process can also be summarized as two basic principles from the macroscopical point of view (Li, Wei & Liu, 2004): the first is that **humans always cognize things of the same kind**, and the second is that **humans recognize and accept things of a new kind easily**.

In order to clarify the idea, the function imitation explanation of NN and SVM is first analyzed. The function imitation of human cognition process for pattern classification (Li & Wei, 2005) via NN and SVM can be explained as follows:

The training process of these machines actually imitates the learning processes of human being, which is called "cognizing process", While the testing process of an unknown sample actually imitates the recognizing process of human being, which is called "recognizing process".

From a mathematical point of view, the feature space is divided into many partitions according to the selected training principles (Haykin, 1994). Each feature space partition is then linked with a corresponding class. Given an unknown sample, NN or SVM detects its position and then assigns the indicator. For more details, the reader should refer to (Haykin, 1994 & Vapnik, 1995).

Now, suppose a sample database is given, if a totally unknown new sample comes, both SVM and NN will not naturally recognize it correctly and will consider it to the closest known in the learned classes (Li & Wei, 2005).

The root cause of this phenomenon lies in the fact that the learning principle of the NN or SVM is based on

feature space partition. This kind of learning principle may amplify each class's distribution region especially when the samples of different kinds are small due to incompleteness. Thus it is impossible for NN or SVM to detect the unknown new samples successfully.

However, this phenomenon is quite easy for humans to handle. Suppose that we have learned some things of the same kind before, and then if given similar things we can easily recognize them. And if we have never encountered with them, we can also easily tell that they are fresh things. Then we can remember their features in the brain. Sure, this process will not affect other learned things. This point surely makes our new learning paradigm different from NN or SVM.

MAIN FOCUS

Humans generally cognize things of one kind and recognize totally unknown things of a new kind easily. So why not let the learning machine "cognize" or "recognize" like humans (Li, Wei & Liu, 2004). Thus our intention is only focused on learning each single class instead of separating them. To learn each class, we can **let each class be cognized or described by a cognitive learner** to imitate the "cognizing process". Therefore, the false alarm zone is now greatly cut down for small samples case. After training, each minimum volume bounding cognitive learner scatters in the feature space. And all learners' boundaries consist of the whole knowledge to the learned classes. If given an unknown sample, the cognitive recognizer then detects **whether the unknown sample is located inside a cognitive learner's boundary** to imitate the "recognizing process". If the sample is totally new (i.e., none of the trained cognitive learner contains the sample), it can be again described by a new cognitive learner and the new obtained learner can be added to the feature space without affecting others. This concludes the basic process of our proposed enclosing machine learning paradigm (Wei, Li & Li, 2007A).

Mathematic Modeling

In order to make previously mentioned ideas practical, they have to be linked with concrete mathematical models (Wei, Li & Li, 2007A). Actually the first principle can be modeled as a minimum volume enclosing problem. The second principle can be ideally modeled as a point detection problem. Generally, the minimum volume enclosing problem is quite hard to handle samples generated in arbitrary distribution and especially if such the distribution shape might be rather complex to be calculated directly. Therefore, an alternative is to use regular shapes such as sphere, ellipsoid and so on to enclose all samples of the same class with the minimum volume objective (Wei, Löfberg, Feng, Li & Li, 2007). Moreover, the approximation method can be easily formulated and efficiently solved as a convex optimization problem.

Enclosing Machine Learning Concepts

The new learning methodology has three aspects. The first is to learn each class respectively, and it is called cognitive learning. The second is to detect the unknown samples' location and determine their indicator, and it is called cognitive classification. While the third is to conduct a new cognitive learning process, and it is called feedback self-learning. The third is to imitate learning samples from an unknown new kind. The whole process is depicted in Figure 1.

Cognitive Learner: A cognitive learner is defined as the bounding boundary of a minimum volume set which encloses all the given samples. The cognitive learner can be either a sphere or an ellipsoid or their combinations. Figure 2 and Figure 3 depict the examples of sphere learner, ellipsoid learner, and combinational ellipsoid learner in 2D.

We conclude that the basic geometric shapes are the best choices, because they have many commendable features: (1) regular to make calculation easier. (2) convex bodies, which guarantee the optimality. (3) fault tolerance to assure generalization performance. And thus operations like intersection, union or complement of them can be implemented easily within convex optimization framework. Then the volume of them can be minimized to enclose the given samples. This is the most important reason why we call it enclosing machine learning.

Remarks: As for the illustrated three type learner, obviously the sphere learner has the biggest volume. Next is single ellipsoid learner. The combinational ellipsoid learner has the smallest volume.

Figure 1. Enclosing machine learning process: The real line denotes the cognizing process. The dotted line denotes the recognizing process. The dash-dotted line denotes the feedback self-learning process.

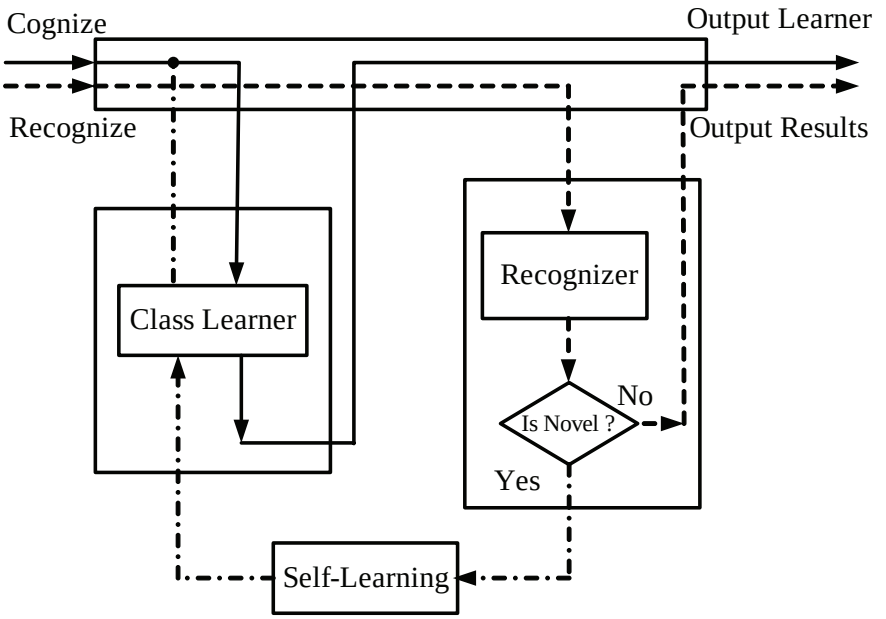


Figure 2. Single cognitive learner illustrations: (a) A sphere learner. (b) An ellipsoid learner

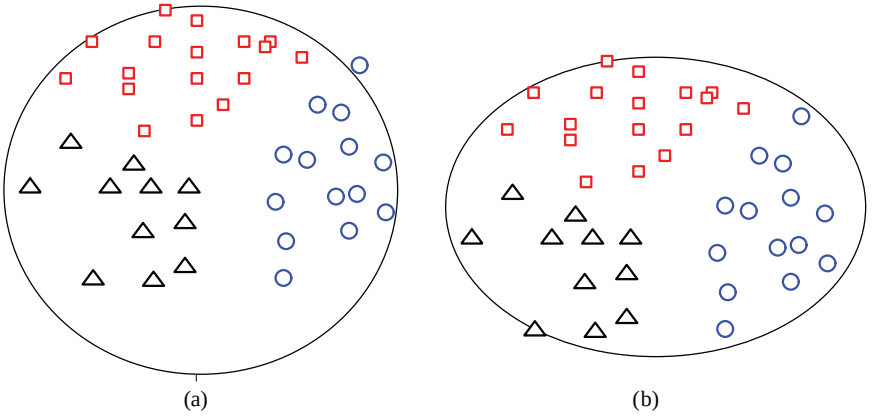
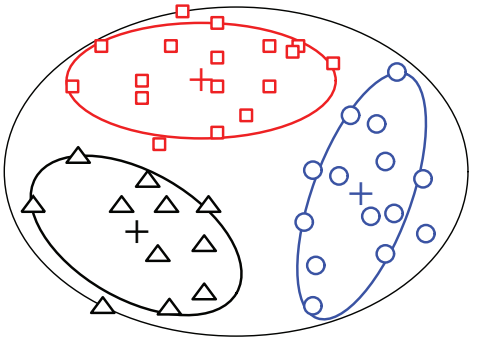


Figure 3. A combinational ellipsoid learner illustration



Cognitive Recognizer: A cognitive recognizer is defined as the point detection and assignment algorithm. For example, it is a distance metric with true or false output.

Cognitive Learning and Classification Algorithms

Firstly, the differences between enclosing machine learning and other feature space partition based methods are investigated. Figure 4 gives a geometric illustration of the differences. As for enclosing machine learning, each class is described by a cognitive learner in the cognitive learning process. But for the partition based learning paradigm such as SVM, every two classes are separated by a hyperplane (or other boundary forms, such as hypersphere etc.). Obviously, the partition based learning paradigm amplifies the real distribution regions of each class, while the enclosing machine learning paradigm obtains more reasonable approximated distribution regions.

Figure 4. A geometric illustration of learning a three class samples via enclosing machine learning vs. feature space partition learning paradigm. (a) For the depicted example, the cognitive learner is the bounding minimum volume ellipsoid, while the cognitive recognizer is actually the point location detection algorithm of the testing sample. (b) All the three classes are separated by three hyperplanes.

In enclosing machine learning, the most important step is to obtain a proper description of each single class. From a mathematical point of view, our cognitive learning methods actually are the same as the so-called one class classification methods (OCCs) (Schölkopf, Platt, Shawe-Taylor, 2001). It is reported that OCCs can efficiently recognize the new samples that resemble the training set and detect uncharacteristic samples, or outliers, which justifies feasibility of our initial idea of the cognizing imitation method.

By far, the well-known examples of OCCs are studied in the context of SVM. Support vector domain description (SVDD) proposed by Tax & Duin (1999) tries to seek the minimum hypersphere that encloses all the data of the target class in a feature space.

However, traditional Euclidean distance based OCCs are often sub-optimal and scale variant. To overcome this, Tax & Juszczak (2003) propose a KPCA based technique to rescale the data in a kernel feature space to unit variance. Except this, Mahalanobis distance based OCCs (Tsang, Kwok, & Li, S., 2006) are reported scale invariant and more compact than traditional ones. What's more, to alleviate the undesirable effects of estimation error, a priori knowledge with an uncertainty model can be easily incorporated.

Utilizing virtues of Mahalanobis distance, our progresses towards "cognizing" are that a new minimum volume enclosing ellipsoid learner is developed and several Mahalanobis distance based OCCs are proposed, i.e. a dual QP ellipsoidal learner (Wei, Huang & Li, 2007A), a dual QP hyperplane learner (Wei, Huang & Li, 2007B), and a primal second order cone programming representable ellipsoidal learner (Wei, Li, Feng & Huang, 2007A). According to the new learners, several practical learning algorithms are developed.

Minimum Volume Enclosing Ellipsoid (MVEE) Learner

Towards the state-of-the-arts MVEE algorithms, they include solving the SDP primal, solving the lndet dual using interior point algorithm and solving the SOCP primal (Wei, Li, Feng, & Huang, 2007B). The minimum volume enclosing ellipsoid center at the origin is only considered for simplicity. As for this point, the reader may check the paper (Wei, Li, Feng & Huang, 2007A) for more details.

Given sample $X \in R^{m \times n}$, suppose $\mathrm{E}(c,\Sigma) := \{x : (x-c)^T \Sigma^{-1} (x-c) \le 1\}$ is the demanded ellipsoid, then the minimum volume enclosing ellipsoid problem can be formulated as

$$\min_{A,b} -\ln\det\Sigma^{-1}$$
$$s.t. \begin{cases} (x_i - c)^T \Sigma^{-1} (x_i - c) \le 1 \\ \Sigma^{-1} \succ 0 \end{cases} \quad (1)$$

However, this is not a convex optimization problem. Fortunately, it can be transformed into the following convex optimization problem:

$$\min_{A,b} -\ln\det A$$
$$s.t. \begin{cases} (Ax_i - b)^T (Ax_i - b) \le 1 \\ A \succ 0, \forall i = 1,2,\cdots,n \end{cases} \quad (2)$$

via matrix transform $\begin{cases} A = \Sigma^{-\frac{1}{2}} \\ b = \Sigma^{-\frac{1}{2}} c \end{cases}$.

In order to allow errors, (2) can be represented in following SDP form

$$\begin{aligned} &\min_{A,b,\zeta_i} -\ln\det A + \Theta\sum_{i=1}^{n}\zeta_i \\ &s.t.\begin{bmatrix} I & (Ax_i - b) \\ (Ax_i - b)^T & 1+\zeta_i \end{bmatrix} \geq 0 \end{aligned} \tag{3}$$

Solving (3), the minimum volume enclosing ellipsoid is then obtained. Yet, SDP is quite demanded especially for large scale or high dimensional data learning problem.

As for $\mathrm{E}(c,\Sigma) := \{x : (x-c)^T \Sigma^{-1} (x-c) \leq R^2\}$, the primal of minimum volume enclosing ellipsoid problem can be reformulated as following SOCP form:

$$\begin{aligned} &\min_{A,b,\zeta_i} -\ln\det A + \Theta\sum_{i=1}^{n}\zeta_i \\ &s.t.\begin{bmatrix} I & (Ax_i - b) \\ (Ax_i - b)^T & 1+\zeta_i \end{bmatrix} \geq 0 \end{aligned} \tag{4}$$

Accordingly, it can be kernelized as:

$$\begin{aligned} &\min_{\mathbf{w},R,\xi_i \geq 0} R + \Theta\sum_{i=1}^{n}\xi_i \\ &s.t.\begin{cases} \sqrt{n}\left\|\mathbf{\Omega}^{-1}\mathbf{Q}(\mathbf{k} - \mathbf{K}\mathbf{w})\right\|_2 \leq R + \xi_i, \\ R > 0, \xi_i \geq 0, i = 1,2,\cdots,n. \end{cases} \end{aligned} \tag{5}$$

where **c** is the center of the ellipsoid, R is the generalized radius, n is number of samples, and $\mathbf{K}_C = \mathbf{Q}^T \Omega \mathbf{Q}$, ξ_i is slack variable, Θ is tradeoff between volume and errors, Σ is covariance matrix.

Except previously mentioned primal based methods, the minimum volume enclosing ellipsoid centered at the origin can also be reformulated as following:

$$\begin{aligned} &\min_{U,\zeta_i} -\ln\det\Sigma^{-1} + \Theta\sum_{i=1}^{n}\zeta_i \\ &s.t.\begin{cases} x_i^T\Sigma^{-1}x_i \leq 1+\zeta_i \\ \zeta_i \geq 0, \forall i = 1,2,\cdots,n \end{cases} \end{aligned} \tag{4}$$

Via optimality and KKT conditions, the dual as following can be efficiently solved

$$\begin{aligned} &\max_{\alpha} \ln\det\sum_{i=1}^{n}\alpha_i x_i x_i^T - \sum_{i=1}^{n}\alpha_i \\ &s.t.\begin{cases} 0 \leq \alpha_i \leq \Theta \\ i = 1,\cdots,n \end{cases} \end{aligned} \tag{5}$$

where α is dual variable.

See that (5) cannot be kernelized directly, therefore it is necessary to use some tricks (Wei, Li, & Dong, 2007) to get the kernelized version:

$$\begin{aligned} &\max_{\alpha} \ln\det \mathrm{K}^{\frac{1}{2}}\Gamma\mathrm{K}^{\frac{1}{2}} - \sum_{i=1}^{n}\alpha_i \\ &s.t.\begin{cases} 0 \leq \alpha_i \leq \Theta \\ i = 1,\cdots,n \end{cases} \end{aligned} \tag{6}$$

where α is the dual variable, $\Gamma := \begin{bmatrix} \alpha_1 & & \\ & \ddots & \\ & & \alpha_n \end{bmatrix}$,

$$\mathrm{K} := \begin{bmatrix} k(x_1,x_1) & \cdots & k(x_1,x_n) \\ \vdots & \ddots & \vdots \\ k(x_n,x_1) & \cdots & k(x_n,x_n) \end{bmatrix}$$

Multiple Class Classification Algorithms

For a multiple class classification problem, a naturally solution is first to use minimum volume geometric shapes to approximate each class samples' distribution separately. But this is for ideal case, where no overlaps occur. When overlaps occur, two multiple class classification algorithms are proposed to handle this case (Wei, Huang & Li, 2007C).

For the first algorithm, a distance based metric is adopted, i.e. assign it to the closest class. This multiple class classification algorithm can be summarized as:

$$f(x) = \arg \min_{k\in\{1,2,\cdots,m\}} \|x-c\|_{\Sigma} - R \tag{7}$$

where $\|\bullet\|_{\Sigma}$ denotes Mahalanobis norm.

Another multiple class classification algorithm is to use optimum Bayesian decision theory, and assign it to the class with maximum posterior probability:

$$f(x) = \arg \max_{k\in\{1,2,\cdots,m\}} \frac{P_k}{(2\pi R_k^2)^{\frac{d}{2}}} \exp(-\frac{\|\mathbf{x}-\mathbf{c}_k\|_M^2}{R_k^2}) \tag{8}$$

where d is the dimension of the feature space and $P_k = \frac{1}{N}\sum_{i=1}^{N} 1_{y_i=k}$ is the prior distribution of class k.

According to (8) the decision boundary between class 1 and 2 is given by

$$\frac{P_1(2\pi R_1^2)^{-\frac{d}{2}} \exp(-\frac{\|\mathbf{x}-\mathbf{c}_1\|_M^2}{R_1^2})}{P_2(2\pi R_2^2)^{-\frac{d}{2}} \exp(-\frac{\|\mathbf{x}-\mathbf{c}_2\|_M^2}{R_2^2})} = 1 \tag{9}$$

And this is equivalent to

$$\frac{\|\mathbf{x}-\mathbf{c}_1\|_M^2}{R_1^2} + T_1 = \frac{\|\mathbf{x}-\mathbf{c}_2\|_M^2}{R_2^2} + T_2 \tag{10}$$

Therefore a new decision rule is given by

$$f(x) = \arg \max(\frac{\|\mathbf{x}-\mathbf{c}_k\|_M^2}{R_k^2} + T_k) \tag{11}$$

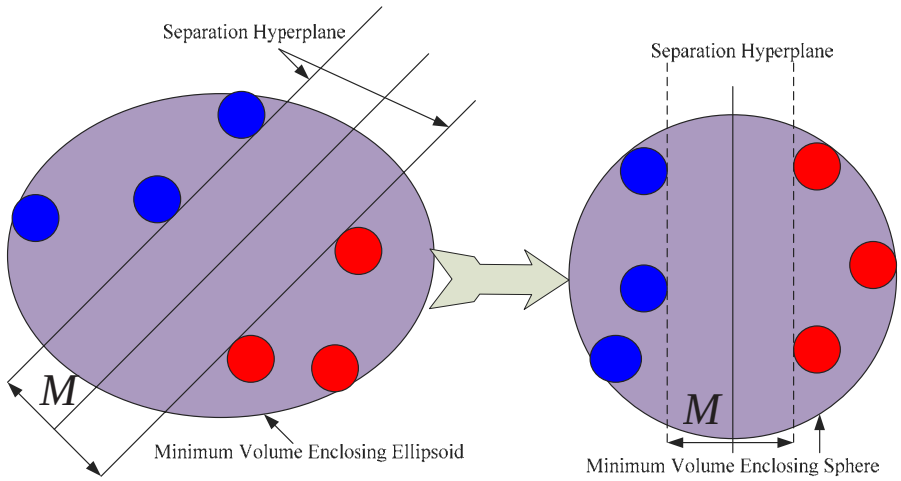
where $T_k = d \log R_k - P_k$ can be estimated from the training samples.

Remarks: A single MVEE learner based two-class classification algorithm (Wei, Li, Feng & Huang, 2007A) is also proposed, which owns both features of MVEE description and SVM discrimination. Then via one-vs-one or one-against-one strategy, a multiple class classification algorithm can be extended. Except this, we recently get an idea of solving a multiple class classification algorithm learning at complexity of a two-class classification algorithm, which is expected to obtain promising performances.

Gap Tolerant SVM Design

In this section, a new gap tolerant SVM design algorithm based on minimum volume enclosing ellipsoid is briefly reviewed. We first find the MVEE around all the samples and thus obtain a Mahalanobis transform. We then use the Mahalanobis transform to whiten all the given samples and map them to a unit sphere distribution (see Figure 5). Then we construct standard SVM there.

Figure 5. MVEE gap tolerant classifier illustration



The MVEE gap tolerant classifier design algorithm can be summarized as:

Step1: Solve MVEE and obtain Σ and center c

Step2: Whiten data using Mahalanobis transformand $t_i = \Sigma^{-\frac{1}{2}}(x_i - c)$ get new sample pairs $(t_i, y_i)_{i=1}^{n}$

Step3: Train standard SVM and get the decision function $y(x) = \text{sgn}(w^T t + b)$.

Remarks: This algorithm is very concise and has several commendable features worth noted: (1) less VC dimension compared with traditional ones. (2) scale invariant. For more details, the reader should refer to (Wei, Li & Dong, 2007).

FUTURE TRENDS

In the future, more basic learner algorithms such as box, convex hull etc, and more compact learner such as set based combinational learner algorithm (Wei, & Li, 2007; Wei, Li, & Li, 2007B) will be developed.

More classification algorithms will be focused, such as vector value reproducing kernel Hilbert space based one. This new algorithm is expected to solve multiple class classification problems efficiently while with single class complexity.

Another topic is how to scale up to large problems. The coreset idea will be introduced for speedup.

Except theoretical developments, various applications such as novelty detection, face detection, and many other possible applications are also expected.

CONCLUSION

In this article, the enclosing machine learning paradigm is introduced. Its concept definitions and progresses in modeling the cognizing process via minimum volume enclosing ellipsoid are reported. Then, several learning and classification algorithms based on MVEE are introduced. And a new gap tolerant SVM design method based MVEE is also reported.

REFERENCES

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*, 1st edn, Oxford: Oxford University Press

Gerstner, W., & Kistler, W. M. (2002). *Spiking Neuron Models: Single Neurons, Populations, Plasticity*, 1st edn, Cambridge: Cambridge University Press.

Haykin, S. (1994). *Neural Networks: A Comprehensive Foundation*, 1st edn, NJ: Prentice Hall Press.

Li, Y. H., & Wei, X. K. (2005). Fusion development of support vector machines and neural networks, *Journal of Air Force Engineering University*, 4, 70-73.

Li, Y. H., Wei, X. K & Liu, J. X. (2004). *Engineering Applications of Support Vector Machines*. 1st edn. Beijing: Weapon Industry Press.

Schölkopf, B., Platt, J., Shawe-Taylor, J., Smola, A., & Williamson, R. (2001). Estimating the support of a high dimensional distribution, *Neural Computation, 13*(7), 1443-1471.

Tax, D. M. J., & Duin, R. P. W. (1999). Support vector domain description, *Pattern Recognition Letters*, 20, 1191-1199.

Tax, D. M. J., & Juszczak, P. (2003) .Kernel whitening for one-class classification, *International Journal of Pattern Recognition and Artificial Intelligence*, *17*(3), 333-347.

Tsang, Ivor W. Kwok, James T. & Li, S. (2006). Learning the kernel in Mahalanobis one-class support vector machines, in *International Joint Conference on Neural Networks 2006*, 1169-1175.

Vapnik, V. N. (1995). *The Nature of Statistical learning theory*, 1st edn, New York: Springer-Verlag.

Wei, X. K., Huang, G. B. & Li, Y. H. (2007A). Mahalanobis ellipsoidal learning machine for one class classification, in *2007 International Conference on Machine Learning and Cybernetics***,** 3528-3533.

Wei, X. K., Huang, G. B. & Li, Y. H. (2007B). A new one class Mahalanobis hyperplane learning machine based on QP and SVD, *Dynamics of Continuous, Discrete and Impulsive Systems Series B: Applications & Algorithms*.

Wei, X. K., Huang, G. B. & Li, Y. H. (2007C). Bayes cognitive ellipsoidal learning machine for recognizing process imitation, *Dynamics of Continuous, Discrete and Impulsive Systems Series B: Applications & Algorithms.*

Wei, X. K., Li, Y. H & Feng, Y. (2006). Comparative study of extreme learning machine and support vector machines. In Wang, J. et al. (Eds.), *International Symposium on Neural Networks 2006*, LNCS 3971, 1089-1095.

Wei, X. K., Li, Y. H., Feng, Y. & Huang, G. B. (2007A). Minimum Mahalanobis enclosing ellipsoid machine for pattern classification, in Huang, D.-S., Heutte, L. & Loog, M. (Eds.), *2007 International Conference on Intelligent Computing*, CCIS 2, 1176–1185.

Wei, X. K., Li, Y. H, Feng, Y. & Huang, G. B. (2007B). Solving Mahalanobis ellipsoidal learning machine via second order cone programming, in Huang, D.-S., Heutte, L. & Loog, M. (Eds.), *2007 International Conference on Intelligent Computing*, CCIS 2, 1186–1194.

Wei, X. K. & Li, Y. H. (2007). Linear programming minimum sphere set covering for extreme learning machines, *Neurocomputing*, 71(4-6), 570-575.

Wei, X. K., Li, Y. H. & Dong, Y. (2007). A new gap tolerant SVM classifier design based on minimum volume enclosing ellipsoid, in *Chinese Conference on Pattern Recognition 2007*.

Wei, X. K., Li, Y. H, & Li, Y. F. (2007A). Enclosing machine learning: concepts and algorithms, *Neural Computing and Applications*, 17(3), 237-243.

Wei, X. K., Li, Y. H. & Li, Y. F. (2007B). Optimum neural network construction via linear programming minimum sphere set covering, in Alhajj, R. et al. (Eds.), *The International Conference on Advanced Data Mining and Applications 2007*, LNAI 4632, 422–429.

Wei, X. K., Löfberg, J., Feng, Y., Li, Y. H., & Li, Y.F. (2007). Enclosing machine learning for class description, In Liu, D. et al. (Eds.), *International Symposium on Neural Networks 2007*, LNCS 4491, 424–433.

E

KEY TERMS

Cognitive Learner: It is defined as the bounding boundary of a minimum volume set which encloses all the given samples to imitate the learning process.

Cognition Process: In this chapter, it refers to the cognizing and recognizing process of the human brain.

Cognitive Recognizer: It is defined as the point detection and assignment algorithm to imitate the recognizing process.

Enclosing Machine Learning: It is a new machine learning paradigm which is based on function imitation of human being's cognizing and recognizing process using minimum enclosing set approximation.

Minimum Enclosing Set: It is a bounding boundary with minimum volume, which encloses all the given points exactly.

MVEE: It is an ellipsoidal minimum enclosing set, which encloses all the given points with minimum volume.

MVEE Gap Tolerant Classifier: A MVEE Gap Tolerant Classifier is specified by an ellipsoid, and by two hyperplanes, with parallel normals. The set of points lying in between (but not on) the hyperplanes is called the margin set. Points that lie inside the ellipsoid but not in the margin set are assigned a class, $\{\pm 1\}$, depending on which side of the margin set they lie on. All other points are defined to be correct: they are not assigned a class. A MVEE gap tolerant classifier is in fact a special kind of support vector machine which does not count data falling outside the ellipsoid containing the training data or inside the margin as an error.

Enhancing Web Search through Query Expansion

Daniel Crabtree
Victoria University of Wellington, New Zealand

INTRODUCTION

Web search engines help users find relevant web pages by returning a result set containing the pages that best match the user's query. When the identified pages have low relevance, the query must be refined to capture the search goal more effectively. However, finding appropriate refinement terms is difficult and time consuming for users, so researchers developed query expansion approaches to identify refinement terms automatically.

There are two broad approaches to query expansion, automatic query expansion (AQE) and interactive query expansion (IQE) (Ruthven et al., 2003). AQE has no user involvement, which is simpler for the user, but limits its performance. IQE has user involvement, which is more complex for the user, but means it can tackle more problems such as ambiguous queries.

Searches fail by finding too many irrelevant pages (low precision) or by finding too few relevant pages (low recall). AQE has a long history in the field of information retrieval, where the focus has been on improving recall (Velez et al., 1997). Unfortunately, AQE often decreased precision as the terms used to expand a query often changed the query's meaning (Croft and Harper (1979) identified this effect and named it query drift). The problem is that users typically consider just the first few results (Jansen et al., 2005), which makes precision vital to web search performance. In contrast, IQE has historically balanced precision and recall, leading to an earlier uptake within web search. However, like AQE, the precision of IQE approaches needs improvement. Most recently, approaches have started to improve precision by incorporating semantic knowledge.

BACKGROUND

While AQE and IQE use distinctly different user interfaces, they use remarkably similar components. Both involve three components: the retrieval model, term weighting, and term selection. The four main retrieval models are the Boolean model, the vector space model, the probabilistic model, and the logical model (Ruthven et al., 2003). Term weighting is dependent on the retrieval model. For the Boolean model, the selected terms simply extend the query. For the vector space model, Rocchio (1971) developed the most common weighting scheme, which increases the weight of relevant terms and decreases the weight of irrelevant terms. The remaining models use richer weighting schemes.

Historically, query expansion approaches targeted specific retrieval models and focused on optimizing the model parameters and the selection of term weights. However, these issues have largely been resolved. The best approaches add a small subset of the most discriminating expansion terms. For web-scale data sets (those involving billions of pages), selecting about 25 terms performs best and selecting more terms decreases precision (Yue et al., 2005). Once selected, the terms are relatively easy to incorporate into any retrieval model and weighting scheme. Therefore, our focus lies on term selection.

The selected terms must address a variety of search problems. A well-known problem is low recall, typically caused by the vocabulary gap (Smyth, 2007), which occurs when some relevant pages do not contain the query terms. For example, a search for "data mining algorithms" may not find relevant pages such as one that discussed "decision trees", which used the term "machine learning" instead of "data mining". Fortunately, AQE and IQE approaches adequately address low recall for web search when the precision of the highest ranked pages is high. That makes precision improvement paramount and the focus of both recent research and this chapter.

Another well-known problem is query ambiguity, which hurts precision and arises when there are multiple interpretations for a query. For example, "jaguar" may refer to the car or the animal. As the user must clarify

their intent, AQE approaches cannot help refine these queries, but for simple cases, many IQE approaches work well. The trouble is distinguishing interpretations that use very similar vocabulary.

A recent TREC workshop identified several problems that affect precision, half involved the aspect coverage problem (Carmel et al., 2006) and the remainder dealt with more complex natural language issues. The aspect coverage problem occurs when pages do not adequately represent all query aspects. For example, a search for "black bear attacks" may find many irrelevant pages that describe the habitat, diet, and features of black bears, but which only mention in passing that sometimes bears attack – in this case, the result set underrepresents the attacks aspect. Interestingly, addressing the aspect coverage problem requires no user involvement and the solution provides insights into distinguishing interpretations that use similar vocabulary.

Query ambiguity and the aspect coverage problem are the major causes of low precision. Figure 1 shows four queries, "regular expressions" (an easy single-aspect query that is not ambiguous), "jaguar" (an easy single-aspect query that is ambiguous), "transportation tunnel disasters" (a hard three-aspect query that is not ambiguous), and "black bear attacks" (a hard two-aspect query that is ambiguous). Current search engines easily solve easy, non-ambiguous queries like "regular expressions" and most IQE approaches address easy, ambiguous queries like "jaguar". The hard queries like "transportation tunnel disasters" are much harder to refine, especially when they are also ambiguous like "black bear attacks".

Predicting query difficulty has been a side challenge for improving query expansion. Yom-Tov et al. (2005) proposed that AQE performance would improve by only refining queries where the result set has high precision. The problem is that users gain little benefit from marginal improvements to easy queries that already have good performance. Therefore, while predicting query difficulty is useful, the challenge is still to improve the precision of the hard queries with low initial precision. Fortunately, the most recent query expansion approaches both improve the precision of hard queries and help predict query difficulty.

MAIN FOCUS

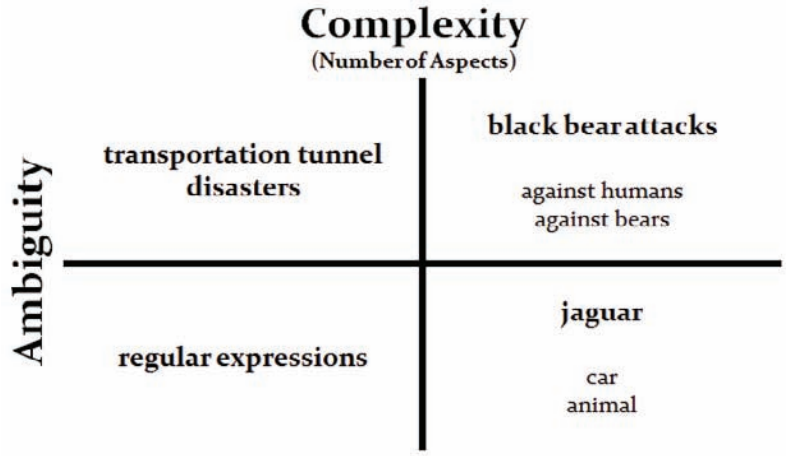
There are a variety of approaches for finding expansion terms, each with its own advantages and disadvantages.

Relevance Feedback (IQE)

Relevance feedback methods select expansion terms based on the user's relevance judgments for a subset of the retrieved pages. Relevance feedback implementations vary according to the use of irrelevant pages and the method of ranking terms.

Normally, users explicitly identify only relevant pages; Ide (1971) suggested two approaches for de-

Figure 1. Query examples and factors affecting precision



termining the irrelevant ones. Ide-regular considers pages irrelevant when they are not relevant and occur earlier in the result set than the last relevant page. For example, when pages four and five are relevant, pages one, two, and three are irrelevant, and positions six and later are neither relevant nor irrelevant. Ide-dec-hi considers just the first non-relevant page irrelevant. While not outperforming Ide-regular, Ide-dec-hi is more consistent and improves more queries (Ruthven et al., 2003).

The most important factor for performance is selecting the most valuable terms. Researchers considered many approaches, including all combinations of document frequency (DF), term frequency (TF), and inverse document frequency (IDF). Experiments consistently confirm that TF•IDF performs the best (Drucker et al., 2002).

Relevance feedback approaches frequently improve query performance when at least some of the initially retrieved pages are relevant, but place high demands on the user. Users must waste time checking the relevance of irrelevant documents and performance is highly dependent on judgment quality. In fact, Magennis & Rijsbergen (1997) found that while user selections improve performance, they were not optimal and often fell short of completely automatic expansion techniques such as pseudo-relevance feedback.

Pseudo-Relevance Feedback (AQE+IQE)

Pseudo-Relevance Feedback, also called blind or ad-hoc relevance feedback is nearly identical to relevance feedback. The difference is that instead of users making relevance judgments, the method assumes the first N pages are relevant. Using approximately N=10 provides the best performance (Shipeng et al., 2003).

Pseudo-relevance feedback works well for AQE when the query already produces good results (high precision). However, for the hard queries, with poor initial results (low precision), pseudo-relevance feedback often reduces precision due to query drift. Crabtree et al. (2007) recently confirmed the problem and found that on balance, pseudo-relevance feedback does not provide any significant improvement over current search engines.

Pseudo-relevance feedback can be adapted for IQE by presenting users with the top ranked terms, and letting them select which terms to use for query expansion. Like relevance feedback, this relies on user judgments, but the user's decision is simpler and performance is potentially greater: Crabtree et al. (2007) found that adding the single best refinement term suggested by interactive pseudo-relevance feedback outperformed relevance feedback when given optimal relevance judgments for five pages.

Thesauri Based Expansion (AQE+IQE)

Thesauri provide semantically similar terms and the approaches based on them expand the query with terms closely related to existing query terms. Human constructed thesauri like WordNet expose an assortment of term relationships. Since many relationships are unsuitable for direct query expansion, most approaches use only a few relationships such as synonyms and hyponyms. Hsu et al. (2006) changed this, by combining several relationships together into a single model for AQE. As with pseudo-relevance feedback, thesauri approaches can present users with thesauri terms for IQE (Sihvonen et al., 2004).

A useful alternative to human constructed thesauri is global document analysis, which can provide automatic measures of term relatedness. Cilibrasi and Vitanyi (2007) introduced a particularly useful measure called Google distance that uses the co-occurrence frequency of terms in web pages to gauge term relatedness.

Human constructed thesauri are most helpful for exploring related queries interactively, as knowledge of term relationships is invaluable for specializing and generalizing refinements. However, descriptively related terms such as synonyms perform poorly for query expansion, as they are often polysemous and cause query drift. Co-occurrence information is more valuable, as good refinements often involve descriptively obscure, but frequently co-occurring terms (Crabtree et al., 2007). However, simply adding co-occurring terms en-mass causes query drift. The real advantage comes when coupling co-occurrence relatedness with a deeper analysis of the query or result set.

Web Page Clustering (IQE)

Web page clustering approaches group similar documents together into clusters. Researchers have developed many clustering algorithms: hierarchical (agglomerative and divisive), partitioning (probabilistic, k-means), grid-based and graph-based clustering (Berkhin, 2002). For web page clustering, algorithms

extend these by considering text or web specific characteristics. For example, Suffix Tree Clustering (Zamir and Etzioni, 1998) and Lingo (Osinski et al., 2004) use phrase information and Menczer (2004) developed an approach that uses the hyperlinked nature of web pages. The most recent trend has been incorporating semantic knowledge. Query Directed Clustering (Crabtree et al., 2006) uses Google distance and other co-occurrence based measures to guide the construction of high quality clusters that are semantically meaningful.

Normally, clustering algorithms refine queries indirectly by presenting the pages within the cluster. However, they can just as easily generate terms for query expansion. Many text-oriented clustering algorithms construct intentional descriptions of clusters comprised of terms that are ideal for query expansion. For the remainder, relevance feedback applies at the cluster level, for instance, by treating all pages within the selected cluster as relevant. Query Directed Clustering additionally computes cluster-page relevance; a particularly nice property for weighting page terms when used for relevance feedback.

Web page clustering enables users to provide feedback regarding many pages simultaneously, which reduces the likelihood of users making relevance judgment mistakes. Additionally, it alleviates the need for users to waste time examining irrelevant pages. The performance is also superior to the alternatives: Crabtree et al. (2007) showed that the best clustering algorithms (Lingo and Query Directed Clustering) outperform the relevance feedback and interactive pseudo-relevance approaches. However, clustering approaches do not adequately address queries where the vocabulary of the relevant and irrelevant documents is very similar, although they do significantly outperform the alternatives discussed previously (Crabtree et al., 2007a).

Query Aspect Approaches (AQE+IQE)

Query aspect approaches (Crabtree et al., 2007) use an in-depth analysis of the query to guide the selection of expansion terms. Broadly, the approach identifies query aspects, determines how well the result set represents each aspect, and finally, finds expansion terms that increase the representation of any underrepresented aspects.

Query aspect approaches identify query aspects by considering the query word ordering and the relative frequencies of the query subsequences; this works as queries with the same intent typically preserve the ordering of words within an aspect. For example, a user wanting "Air New Zealand" would not enter "Zealand Air New". Determining aspect representation involves examining the aspect's vocabulary model, which consists of terms that frequently co-occur with the aspect. The degree of aspect representation is roughly the number of terms from the aspect's vocabulary model in the result set; this works as different aspects invoke different terms. For example, for the query "black bear attacks", the "black bear" aspect will invoke terms like "Animal", "Mammal", and "Diet", but the "attacks" aspect will invoke terms like "Fatal", "Maul", and "Danger". The terms selected for query expansion are terms from the vocabulary models of underrepresented aspects that resolve the underrepresentation of query aspects.

Query aspects help predict query difficulty: when the result set underrepresents an aspect, the query will have poor precision. Additionally, query aspect approaches can improve exactly those queries that need the most improvement. AbraQ (Crabtree et al., 2007) provides a very efficient approach that solves the aspect coverage problem by using the query aspect approach for AQE. AbraQ without any user involvement produces higher precision queries than the alternatives given optimal user input. Qasp (Crabtree et al., 2007a) coupled the query aspect approach with hierarchical agglomerative clustering to provide an IQE approach that works when the vocabulary of the relevant and irrelevant documents is very similar. Furthermore, with a trivial extension, query aspect approaches can improve queries that find few or no results.

Query Log Analysis (IQE)

Query log analysis examines the search patterns of users by considering the pages visited and the sequences of queries performed (Cui, 2002). Query log analysis then presents the user with similar queries as possible expansions. As of October 2007, Google offered a similar queries feature presented as search navigation through the experimental search feature of Google Labs (http://www.google.com/experimental/).

The approach currently works best for short queries repeated by many users (Crabtree et al., 2007). However, there is great potential for cross-fertilization between query log analysis and other query expansion techniques. Unfortunately, this research is constrained

to large search engines, as the query log data has commercial value and is not widely available to other researchers.

FUTURE TRENDS

Over the last few years, the trend has been the increased incorporation of semantic knowledge and the coupling of distinct approaches. These trends will continue into the future and if search companies make at least some query log data publicly available, the coupling between query log analysis and other approaches is likely to be particularly fruitful. Additionally, with the aspect coverage problem addressed, it is likely that the less common natural language problems identified in TREC will become the focus of future efforts.

CONCLUSION

Many approaches select useful terms for query expansion. Relevance feedback and pseudo-relevance feedback help when the initially retrieved documents are reasonably relevant. Thesauri based approaches help users explore the space around their query and boost the performance of other approaches by providing semantic knowledge. Clustering methods help users ignore the irrelevant documents and reduce the amount of feedback required from users. Query aspect approaches identify many problematic queries and improve them without user involvement, additionally query aspect approaches aid clustering by showing how to identify clusters when all documents share similar vocabulary. Query log analysis helps for short and frequent queries, but could potentially improve other approaches in the future.

REFERENCES

Berkhin, P. (2002). *Survey of clustering data mining techniques* (Tech. Rep.). San Jose, CA, USA: Accrue Software.

Buckley, C. (2004). Why current IR engines fail. In *Proceedings of the 27th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 584–585).

Carmel, D., Yom-Tov, E., Darlow, A., & Pelleg, D. (2006). What makes a query difficult? In *Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 390–397).

Cilibrasi, R., L., & Vitanyi, P., M. (2007). The google similarity distance. *IEEE Transactions on Knowledge and Data Engineering*, 19(3), 370-383.

Crabtree, D., Andreae, P., Gao, X. (2006). Query Directed Web Page Clustering. In *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence* (pp. 202-210).

Crabtree, D., Andreae, P., Gao, X. (2007). Exploiting underrepresented query aspects for automatic query expansion. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 191-200).

Crabtree, D., Andreae, P., Gao, X. (2007a). Understanding Query Aspects with applications to Interactive Query Expansion. In *Proceedings of the 2007 IEEE/WIC/ACM International Conference on Web Intelligence* (pp. 691-695).

Croft, W., & Harper, D. (1979). Using probabilistic models of information retrieval without relevance information. *Journal of Documentation*, 35(4), 285-295.

Cui, H., Wen, J., R., Nie, J., Y., & Ma, W., Y. (2002). Probabilistic query expansion using query logs. In *Proceedings of 11th International World Wide Web Conference* (pp. 325-332).

Drucker, H., Shahraray, B., & Gibbon, D. (2002). Support vector machines: relevance feedback and information retrieval. *Information Processing and Management*, 38(3), 305-323.

Hsu, M., H., Tsai, M., F., & Chen, H., H. (2006) Query expansion with ConceptNet and WordNet: An intrinsic comparison. In *Proceedings of Information Retrieval Technology, Third Asia Information Retrieval Symposium* (pp. 1-13).

Ide, E. (1971). New Experiments in relevance feedback. In G. Salton (Ed.), *The SMART retrieval system – experiments in automatic document processing* (pp. 337-354). Prentice-Hall, Inc.

Jansen, B., J., Spink, A., & Pedersen, J., O. (2005). A temporal comparison of AltaVista web searching.

Journal of the American Society for Information Science and Technology, 56(6), 559-570.

Magennis, M., & Rijsbergen, C., J. (1997). The potential and actual effectiveness of interactive query expansion. In *Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 324–332).

Menczer, F. (2004). Lexical and semantic clustering by web links. *Journal of the American Society for Information Science and Technology*, 55(14), 1261-1269.

Osinski, S., Stefanowski, J., & Weiss, D. (2004). Lingo: Search results clustering algorithm based on singular value decomposition. In *Proceedings of the International IIS: Intelligent Information Processing and Web Mining Conference, Advances in Soft Computing* (pp. 359-368).

Rocchio, J., J. (1971). Relevance feedback in information retrieval. In G. Salton (Ed.), *The SMART retrieval system – experiments in automatic document processing* (pp. 313-323). Prentice-Hall, Inc.

Ruthven, I., & Lalmas, M. (2003). A survey on the use of relevance feedback for information access systems. *The Knowledge Engineering Review*, 19(2), 95–145.

Sihvonen, A., & Vakkari, P. (2004). Subject knowledge, thesaurus-assisted query expansion and search success. In *Proceedings of the RIAO 2004 Conference* (pp. 393-404).

Shipeng, Y., Cai, D., Wen, J., & Ma, W. (2003). Improving pseudo-relevance feedback in web information retrieval using web page segmentation. In *Proceedings of the Twelfth International World Wide Web Conference* (pp. 11-18).

Smyth, B. (2007). A Community-Based Approach to Personalizing Web Search. *Computer*, 40(8), 42-50.

Velez, B., Weiss, R., Sheldon, M., A., & Gifford, D., K. (1997). Fast and effective query refinement. In *Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 6-15).

Yom-Tov, E., Fine, S., Carmel, D., & Darlow, A. (2005). Learning to estimate query difficulty: including applications to missing content detection and distributed information retrieval. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 512–519).

Yue, W., Chen, Z., Lu, X., Lin, F., & Liu, J. (2005). Using Query Expansion and Classification for Information Retrieval. In *First International Conference on Semantics, Knowledge and Grid* (pp. 31-37).

Zamir, O., & Etzioni, O. (1998). Web document clustering: A feasibility demonstration. In *Research and Development in Information Retrieval* (pp. 46–54).

KEY TERMS

Automatic Query Expansion: Adding additional query terms to improve result set quality without user involvement.

Interactive Query Expansion: Finding additional query terms to improve result set quality and then presenting them to the user.

Pseudo-relevance Feedback: Selecting refinement terms based on assuming the top N pages are relevant.

Query Aspect: Sequence of query words that form a distinct semantic component of a query. For example, searching for information on holidays has one query aspect (holidays), whereas searching for travel agents in Los Angeles who deal with cruises involves three different query aspects (travel agents, Los Angeles, and cruises).

Query Log Analysis: Mining the collective queries of users for similar queries by finding temporal patterns in the search sequences across user search sessions.

Relevance Feedback: Selecting refinement terms based on the relevancy judgments of pages from the user.

Vocabulary: The terms expected to co-occur in the context of another term.

Web Page Clustering: Automatically grouping together semantically similar pages into clusters by analyzing the contents of pages such as text, page structure, and hyperlinks.

Enhancing Web Search through Query Log Mining

Ji-Rong Wen
Microsoft Research Asia, China

INTRODUCTION

Web query log is a type of file keeping track of the activities of the users who are utilizing a search engine. Compared to traditional information retrieval setting in which documents are the only information source available, query logs are an additional information source in the Web search setting. Based on query logs, a set of Web mining techniques, such as log-based query clustering, log-based query expansion, collaborative filtering and personalized search, could be employed to improve the performance of Web search.

BACKGROUND

Web usage mining is an application of data mining techniques to discovering interesting usage patterns from Web data, in order to understand and better serve the needs of Web-based applications. Since the majority of usage data is stored in Web logs, usage mining is usually also referred to as log mining. Web logs can be divided into three categories based on the location of data collecting: server log, client log, and proxy log. Server log provides an aggregate picture of the usage of a service by all users, while client log provides a complete picture of usage of all services by a particular client, with the proxy log being somewhere in the middle (Srivastava, Cooley, Deshpande, & Tan, 2000).

Query log mining could be viewed as a special kind of Web usage mining. While there is a lot of work about mining Website navigation logs for site monitoring, site adaptation, performance improvement, personalization and business intelligence, there is relatively little work of mining search engines' query logs for improving Web search performance. In early years, researchers have proved that relevance feedback can significantly improve retrieval performance if users provide sufficient and correct relevance judgments for queries (Xu & Croft, 2000). However, in real search scenarios, users are usually reluctant to explicitly give their relevance feedback. A large amount of users' past query sessions have been accumulated in the query logs of search engines. Each query session records a user query and the corresponding pages the user has selected to browse. Therefore, a query log can be viewed as a valuable source containing a large amount of users' implicit relevance judgments. Obviously, these relevance judgments can be used to more accurately detect users' query intentions and improve the ranking of search results.

One important assumption behind query log mining is that the clicked pages are "relevant" to the query. Although the clicking information is not as accurate as explicit relevance judgment in traditional relevance feedback, the user's choice does suggest a certain degree of relevance. In the long run with a large amount of log data, query logs can be treated as a reliable resource containing abundant implicit relevance judgments from a statistical point of view.

MAIN THRUST

Web Query Log Preprocessing

Typically, each record in a Web query log includes the IP address of the client computer, timestamp, the URL of the requested item, the type of Web browser, protocol, etc. The Web log of a search engine records various kinds of user activities, such as submitting queries, clicking URLs in the result list, getting HTML pages and skipping to another result list. Although all these activities reflect, more or less, a user's intention, the query terms and the Web pages the user visited are the most important data for mining tasks. Therefore, a query session, the basic unit of mining tasks, is defined as a query submitted to a search engine together with the Web pages the user visits in response to the query.

Since the HTTP protocol requires a separate connection for every client-server interaction, the activities of multiple users usually interleave with each other. There are no clear boundaries among user query sessions in the logs, which makes it a difficult task to extract individual query sessions from Web query logs (Cooley, Mobasher, & Srivastava, 1999). There are mainly two steps to extract query sessions from query logs: user identification and session identification. User identification is the process of isolating from the logs the activities associated with an individual user. Activities of the same user could be grouped by their IP addresses, agent types, site topologies, cookies, user IDs, etc. The goal of session identification is to divide the queries and page accesses of each user into individual sessions. Finding the beginning of a query session is trivial: a query session begins when a user submits a query to a search engine. However, it is difficult to determine when a search session ends. The simplest method of achieving this is through a timeout, where if the time between page requests exceeds a certain limit, it is assumed that the user is starting a new session.

Log-Based Query Clustering

Query clustering is a technique aiming at grouping users' semantically (not syntactically) related queries in Web query logs. Query clustering could be applied to FAQ detecting, index-term selection and query reformulation, which are effective ways to improve Web search. First of all, FAQ detecting means to detect Frequently Asked Questions (FAQs), which can be achieved by clustering similar queries in the query logs. A cluster being made up of many queries can be considered as a FAQ. Some search engines (e.g. Askjeeves) prepare and check the correct answers for FAQs by human editors, and a significant majority of users' queries can be answered precisely in this way. Second, inconsistency between term usages in queries and those in documents is a well-known problem in information retrieval, and the traditional way of directly extracting index terms from documents will not be effective when the user submits queries containing terms different from those in the documents. Query clustering is a promising technique to provide a solution to the word mismatching problem. If similar queries can be recognized and clustered together, the resulting query clusters will be very good sources for selecting additional index terms for documents. For example, if queries such as "atomic bomb", "Manhattan Project", "Hiroshima bomb" and "nuclear weapon" are put into a query cluster, this cluster, not the individual terms, can be used as a whole to index documents related to atomic bomb. In this way, any queries contained in the cluster can be linked to these documents. Third, most words in the natural language have inherent ambiguity, which makes it quite difficult for user to formulate queries with appropriate words. Obviously, query clustering could be used to suggest a list of alternative terms for users to reformulate queries and thus better represent their information needs.

The key problem underlying query clustering is to determine an adequate similarity function so that truly similar queries can be grouped together. There are mainly two categories of methods to calculate the similarity between queries: one is based on query content, and the other on query session. Since queries with the same or similar search intentions may be represented with different words and the average length of Web queries is very short, content-based query clustering usually does not perform well.

Using query sessions mined from query logs to cluster queries is proved to be a more promising method (Wen, Nie, & Zhang, 2002). Through query sessions, "query clustering" is extended to "query session clustering". The basic assumption here is that the activities following a query are relevant to the query and represent, to some extent, the semantic features of the query. The query text and the activities in a query session as a whole can represent the search intention of the user more precisely. Moreover, the ambiguity of some query terms is eliminated in query sessions. For instance, if a user visited a few tourism Websites after submitting a query "Java", it is reasonable to deduce that the user was searching for information about "Java Island", not "Java programming language" or "Java coffee". Moreover, query clustering and document clustering can be combined and reinforced with each other (Beeferman & Berger, 2000).

Log-Based Query Expansion

Query expansion involves supplementing the original query with additional words and phrases, which is an effective way to overcome the term-mismatching problem and to improve search performance. Log-based query expansion is a new query expansion method based on query log mining. Taking query sessions in query logs

as a bridge between user queries and Web pages, probabilistic correlations between terms in queries and those in pages can then be established. With these term-term correlations, relevant expansion terms can be selected from the documents for a query. For example, a recent work by Cui, Wen, Nie, and Ma (2003) shows that, from query logs, some very good terms, such as "personal computer", "Apple Computer", "CEO", "Macintosh" and "graphical user interface", can be detected to be tightly correlated to the query "Steve Jobs", and using these terms to expand the original query can lead to more relevant pages.

Experiments by Cui, Wen, Nie, and Ma (2003) show that mining user logs is extremely useful for improving retrieval effectiveness, especially for very short queries on the Web. The log-based query expansion overcomes several difficulties of traditional query expansion methods because a large number of user judgments can be extracted from user logs, while eliminating the step of collecting feedbacks from users for ad-hoc queries. Log-based query expansion methods have three other important properties. First, the term correlations are pre-computed offline and thus the performance is better than traditional local analysis methods which need to calculate term correlations on the fly. Second, since user logs contain query sessions from different users, the term correlations can reflect the preference of the majority of the users. Third, the term correlations may evolve along with the accumulation of user logs. Hence, the query expansion process can reflect updated user interests at a specific time.

Collaborative Filtering and Personalized Web Search

Collaborative filtering and personalized Web search are two most successful examples of personalization on the Web. Both of them heavily rely on mining query logs to detect users' preferences or intentions.

Collaborative Filtering

Collaborative filtering is a method of making automatic predictions (filtering) about the interests of a user by collecting taste information from many users (collaborating). The basic assumption of collaborative filtering is that users having similar tastes on some items may also have similar preferences on other items. From Web logs, a collection of items (books, music CDs, movies, etc.) with users' searching, browsing and ranking information could be extracted to train prediction and recommendation models. For a new user with a few items he/she likes or dislikes, other items meeting the user's taste could be selected based on the trained models and recommended to him/her (Shardanand & Maes, 1995; Konstan, Miller, Maltz, Herlocker, & Gordon, L. R., et al., 1997). Generally, vector space models and probabilistic models are two major models for collaborative filtering (Breese, Heckerman, & Kadie, 1998). Collaborative filtering systems have been implemented by several e-commerce sites including Amazon.

Collaborative filtering has two main advantages over content-based recommendation systems. First, content-based annotations of some data types are often not available (such as video and audio) or the available annotations usually do not meet the tastes of different users. Second, through collaborative filtering, multiple users' knowledge and experiences could be remembered, analyzed and shared, which is a characteristic especially useful for improving new users' information seeking experiences.

The original form of collaborative filtering does not use the actual content of the items for recommendation, which may suffers from the scalability, sparsity and synonymy problems. Most importantly, it is incapable of overcoming the so-called first-rater problem. The first-rater problem means that objects newly introduced into the system have not been rated by any users and can therefore not be recommended. Due to the absence of recommendations, users tend to not be interested in these new objects. This in turn has the consequence that the newly added objects remain in their state of not being recommendable. Combining collaborative and content-based filtering is therefore a promising approach to solving the above problems (Baudisch, 1999).

Personalized Web Search

While most modern search engines return identical results to the same query submitted by different users, personalized search targets to return results related to users' preferences, which is a promising way to alleviate the increasing information overload problem on the Web. The core task of personalization is to obtain the preference of each individual user, which is called user profile. User profiles could be explicitly provided

by users or implicitly learned from users' past experiences (Hirsh, Basu, & Davison, 2000; Mobasher, Cooley, & Srivastava, 2000). For example, Google's personalized search requires users to explicitly enter their preferences. In the Web context, query log provides a very good source for learning user profiles since it records the detailed information about users' past search behaviors. The typical method of mining a user's profile from query logs is to first collect all of this user's query sessions from logs, and then learn the user's preferences on various topics based on the queries submitted and the pages viewed by the user (Liu, Yu, & Meng, 2004).

User preferences could be incorporated into search engines to personalize both their relevance ranking and their importance ranking. A general way for personalizing relevance ranking is through query reformulation, such as query expansion and query modification, based on user profiles. A main feature of modern Web search engines is that they assign importance scores to pages through mining the link structures and the importance scores will significantly affect the final ranking of search results. The most famous link analysis algorithms are PageRank (Page, Brin, Motwani, & Winograd, 1998) and HITS (Kleinberg, 1998). One main shortcoming of PageRank is that it assigns a static importance score to a page, no matter who is searching and what query is being used. Recently, a topic-sensitive PageRank algorithm (Haveliwala, 2002) and a personalized PageRank algorithm (Jeh & Widom, 2003) are proposed to calculate different PageRank scores based on different users' preferences and interests. A search engine using topic-sensitive PageRank and Personalized PageRank is expected to retrieve Web pages closer to user preferences.

FUTURE TRENDS

Although query log mining is promising to improve the information seeking process on the Web, the number of publications in this field is relatively small. The main reason might lie in the difficulty of obtaining a large amount of query log data. Due to lack of data, especially lack of a standard data set, it is difficult to repeat and compare existing algorithms. This may block knowledge accumulation in the field. On the other hand, search engine companies are usually reluctant to make pubic of their log data because of the costs and/or legal issues involved. However, to create such a kind of standard log dataset will greatly benefit both the research community and the search engine industry, and thus may become a major challenge of this field in the future.

Nearly all of the current published experiments were conducted on snapshots of small to medium scale query logs. In practice, query logs are usually continuously generated and with huge sizes. To develop scalable and incremental query log mining algorithms capable of handling the huge and growing dataset is another challenge.

We also foresee that client-side query log mining and personalized search will attract more and more attentions from both academy and industry. Personalized information filtering and customization could be effective ways to ease the deteriorated information overload problem.

CONCLUSION

Web query log mining is an application of data mining techniques to discovering interesting knowledge from Web query logs. In recent years, some research work and industrial applications have demonstrated that Web log mining is an effective method to improve the performance of Web search. Web search engines have become the main entries for people to exploit the Web and find needed information. Therefore, to keep improving the performance of search engines and to make them more user-friendly are important and challenging tasks for researchers and developers. Query log mining is expected to play a more and more important role in enhancing Web search.

REFERENCES

Baudisch, P. (1999). Joining collaborative and content-based filtering. *Proceedings of the Conference on Human Factors in Computing Systems (CHI'99).*

Beeferman, D., & Berger. A. (2000). Agglomerative clustering of a search engine query log. *Proceedings*

of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 407-416).

Breese, J., Heckerman, D., & Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence* (pp. 43-52).

Cooley, R., Mobasher, B., & Srivastava, J. (1999). Data preparation for mining World Wide Web browsing patterns. *Journal of Knowledge and Information Systems, 1*(1).

Cui, H., Wen, J.-R., Nie, J.-Y., & Ma, W.-Y. (2003). Query expansion by mining user logs. *IEEE Transaction on Knowledge and Data Engineering, 15*(4), 829-839.

Haveliala, T. H. (2002). Topic-sensitive PageRank. *Proceeding of the Eleventh World Wide Web conference (WWW 2002)*.

Hirsh, H., Basu, C., & Davison, B. D. (2000). Learning to personalize. *Communications of the ACM, 43*(8), 102-106.

Jeh, G., & Widom, J. (2003). Scaling personalized Web search. *Proceedings of the Twelfth International World Wide Web Conference (WWW 2003).*

Kleinberg, J. (1998). Authoritative sources in a hyperlinked environment. *Proceedings of the 9th ACM SIAM International Symposium on Discrete Algorithms* (pp. 668-677).

Konstan, J. A., Miller, B. N., Maltz, D., Herlocker, J. L., Gordon, L. R., & Riedl, J. (1997). GroupLens: applying collaborative filtering to Usenet news. *Communications of the ACM, 40*(3), 77-87.

Liu, F., Yu, C., & Meng, W. (2004). Personalized Web search for improving retrieval effectiveness. *IEEE Transactions on Knowledge and Data Engineering, 16*(1), 28-40.

Mobasher, B., Cooley, R., & Srivastava, J. (2000). Automatic personalization based on Web usage mining. *Communications of the ACM, 43*(8), 142-151.

Page, L., Brin, S., Motwani, R., & Winograd, T. (1998). *The PageRank citation ranking: Bringing order to the Web*. Technical report of Stanford University.

Shardanand, U., & Maes, P. (1995). Social information filtering: Algorithms for automating word of mouth. *Proceedings of the Conference on Human Factors in Computing Systems*.

Srivastava, J., Cooley, R., Deshpande, M., & Tan, P. (2000). Web usage mining: Discovery and applications of usage patterns from Web data. *SIGKDD Explorations, 1*(2), 12-23.

Wen, J.-R., Nie, J.-Y., & Zhang, H.-J. (2002). Query clustering using user logs. *ACM Transactions on Information Systems (ACM TOIS), 20*(1), 59-81.

Xu, J., & Croft, W.B. (2000). Improving the effectiveness of information retrieval with local context analysis. *ACM Transactions on Information Systems, 18*(1), 79-112.

KEY TERMS

Collaborative Filtering: A method of making automatic predictions (filtering) about the interests of a user by collecting taste information from many users (collaborating).

Log-Based Query Clustering: A technique aiming at grouping users' semantically related queries collected in Web query logs.

Log-Based Query Expansion: A new query expansion method based on query log mining. Probabilistic correlations between terms in the user queries and those in the documents can then be established through user logs. With these term-term correlations, relevant expansion terms can be selected from the documents for a query.

Log-Based Personalized Search: Personalized search targets to return results related to users' preferences. The core task of personalization is to obtain the preference of each individual user, which could be learned from query logs.

Query Log: A type of file keeping track of the activities of the users who are utilizing a search engine.

Query Log Mining: An application of data mining techniques to discover interesting knowledge from Web query logs. The mined knowledge is usually used to enhance Web search.

Query Session: a query submitted to a search engine together with the Web pages the user visits in response to the query. Query session is the basic unit of many query log mining tasks.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 438-442, copyright 2005 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

Enhancing Web Search through Web Structure Mining

Ji-Rong Wen
Microsoft Research Asia, China

INTRODUCTION

The Web is an open and free environment for people to publish and get information. Everyone on the Web can be either an author, a reader, or both. The language of the Web, *HTML (Hypertext Markup Language)*, is mainly designed for information display, not for semantic representation. Therefore, current Web search engines usually treat Web pages as unstructured documents, and traditional information retrieval (IR) technologies are employed for Web page parsing, indexing, and searching. The unstructured essence of Web pages seriously blocks more accurate search and advanced applications on the Web. For example, many sites contain structured information about various products. Extracting and integrating product information from multiple Web sites could lead to powerful search functions, such as comparison shopping and business intelligence. However, these structured data are embedded in Web pages, and there are no proper traditional methods to extract and integrate them. Another example is the link structure of the Web. If used properly, information hidden in the links could be taken advantage of to effectively improve search performance and make Web search go beyond traditional information retrieval (Page, Brin, Motwani, & Winograd, 1998, Kleinberg, 1998).

Although *XML (Extensible Markup Language)* is an effort to structuralize Web data by introducing semantics into tags, it is unlikely that common users are willing to compose Web pages using XML due to its complication and the lack of standard schema definitions. Even if XML is extensively adopted, a huge amount of pages are still written in the HTML format and remain unstructured. Web structure mining is the class of methods to automatically discover structured data and information from the Web. Because the Web is dynamic, massive and heterogeneous, automated Web structure mining calls for novel technologies and tools that may take advantage of state-of-the-art technologies from various areas, including machine learning, data mining, information retrieval, and databases and natural language processing.

BACKGROUND

Web structure mining can be further divided into three categories based on the kind of structured data used.

- **Web graph mining:** Compared to a traditional document set in which documents are independent, the Web provides additional information about how different documents are connected to each other via hyperlinks. The Web can be viewed as a (directed) graph whose nodes are the Web pages and whose edges are the hyperlinks between them. There has been a significant body of work on analyzing the properties of the Web graph and mining useful structures from it (Page et al., 1998; Kleinberg, 1998; Bharat & Henzinger, 1998; Gibson, Kleinberg, & Raghavan, 1998). Because the Web graph structure is across multiple Web pages, it is also called *interpage structure*.
- **Web information extraction (Web IE):** In addition, although the documents in a traditional information retrieval setting are treated as plain texts with no or few structures, the content within a Web page does have inherent structures based on the various HTML and XML tags within the page. While Web content mining pays more attention to the content of Web pages, Web information extraction has focused on automatically extracting structures with various accuracy and granularity out of Web pages. Web content structure is a kind of structure embedded in a single Web page and is also called *intrapage structure*.
- **Deep Web mining:** Besides Web pages that are accessible or crawlable by following the hyperlinks, the Web also contains a vast amount of noncrawlable content. This hidden part of the

Web, referred to as the *deep Web* or the *hidden Web* (Florescu, Levy, & Mendelzon, 1998), comprises a large number of online Web databases. Compared to the static surface Web, the deep Web contains a much larger amount of high-quality structured information (Chang, He, Li, & Zhang, 2003). Automatically discovering the structures of Web databases and matching semantically related attributes between them is critical to understanding the structures and semantics of the deep Web sites and to facilitating advanced search and other applications.

MAIN THRUST

Web Graph Mining

Mining the Web graph has attracted a lot of attention in the last decade. Some important algorithms have been proposed and have shown great potential in improving the performance of Web search. Most of these mining algorithms are based on two assumptions. (a) Hyperlinks convey human endorsement. If there exists a link from page A to page B, and these two pages are authored by different people, then the first author found the second page valuable. Thus the importance of a page can be propagated to those pages it links to. (b) Pages that are co-cited by a certain page are likely related to the same topic. Therefore, the popularity or importance of a page is correlated to the number of incoming links to some extendt, and related pages tend to be clustered together through dense linkages among them.

Hub and Authority

In the Web graph, a *hub* is defined as a page containing pointers to many other pages, and an *authority* is defined as a page pointed to by many other pages. An authority is usually viewed as a good page containing useful information about one topic, and a hub is usually a good source to locate information related to one topic. Moreover, a *good* hub should contain pointers to many good authorities, and a *good* authority should be pointed to by many good hubs. Such a mutual reinforcement relationship between hubs and authorities is taken advantage of by an iterative algorithm called HITS (Kleinberg, 1998). HITS computes authority scores and hub scores for Web pages in a subgraph of the Web, which is obtained from the (subset of) search results of a query with some predecessor and successor pages.

Bharat and Henzinger (1998) addressed three problems in the original HITS algorithm: mutually reinforced relationships between hosts (where certain documents "conspire" to dominate the computation), automatically generated links (where no human's opinion is expressed by the link), and irrelevant documents (where the graph contains documents irrelevant to the query topic). They assign each edge of the graph an authority weight and a hub weight to solve the first problem and combine connectivity and content analysis to solve the latter two. Chakrabarti, Joshi, and Tawde (2001) addressed another problem with HITS: regarding the whole page as a hub is not suitable, because a page always contains multiple regions in which the hyperlinks point to different topics. They proposed to disaggregate hubs into coherent regions by segmenting the DOM (document object model) tree of an HTML page.

PageRank

The main drawback of the HITS algorithm is that the hubs and authority score must be computed iteratively from the query result on the fly, which does not meet the real-time constraints of an online search engine. To overcome this difficulty, Page et al. (1998) suggested using a random surfing model to describe the probability that a page is visited and taking the probability as the importance measurement of the page. They approximated this probability with the famous PageRank algorithm, which computes the probability scores in an iterative manner. The main advantage of the PageRank algorithm over the HITS algorithm is that the importance values of all pages are computed off-line and can be directly incorporated into ranking functions of search engines.

Noisy link and topic drifting are two main problems in the classic Web graph mining algorithms. Some links, such as banners, navigation panels, and advertisements, can be viewed as noise with respect to the query topic and do not carry human editorial endorsement. Also, hubs may be mixed, which means that only a portion of the hub content may be relevant to the query. Most link analysis algorithms treat each Web page as an atomic, indivisible unit with no internal structure. This leads to false reinforcements of hub/authority and importance calculation. Cai, He, Wen, and Ma (2004)

used a vision-based page segmentation algorithm to partition each Web page into blocks. By extracting the page-to-block, block-to-page relationships from the link structure and page layout analysis, a semantic graph over the Web can be constructed such that each node exactly represents a single semantic topic. This graph can better describe the semantic structure of the Web. Based on block-level link analysis, they proposed two new algorithms, Block Level PageRank and Block Level HITS, whose performances are shown to exceed the classic PageRank and HITS algorithms.

Community Mining

Many communities, either in an explicit or implicit form, exist in the Web today, and their number is growing at a very fast speed. Discovering communities from a network environment such as the Web has recently become an interesting research problem. The Web can be abstracted into directional or nondirectional graphs with nodes and links. It is usually rather difficult to understand a network's nature directly from its graph structure, particularly when it is a large scale complex graph. Data mining is a method to discover the hidden patterns and knowledge from a huge network. The mined knowledge could provide a higher logical view and more precise insight of the nature of a network and will also dramatically decrease the dimensionality when trying to analyze the structure and evolution of the network.

Quite a lot of work has been done in mining the implicit communities of users, Web pages, or scientific literature from the Web or document citation database using content or link analysis. Several different definitions of community were also raised in the literature. In Gibson et al. (1998), a Web community is a number of representative authority Web pages linked by important hub pages that share a common topic. Kumar, Raghavan, Rajagopalan, and Tomkins (1999) define a Web community as a highly linked bipartite subgraph with at least one core containing complete bipartite subgraph. In Flake, Lawrence, and Lee Giles (2000), a set of Web pages that linked more pages in the community than those outside of the community could be defined as a Web community. Also, a research community could be based on a single-most-cited paper and could contain all papers that cite it (Popescul, Flake, Lawrence, Ungar, & Lee Giles, 2000).

Web Information Extraction

Web IE has the goal of pulling out information from a collection of Web pages and converting it to a homogeneous form that is more readily digested and analyzed for both humans and machines. The results of IE could be used to improve the indexing process, because IE removes irrelevant information in Web pages and facilitates other advanced search functions due to the structured nature of data. The structuralization degrees of Web pages are diverse. Some pages can be just taken as plain text documents. Some pages contain a little loosely structured data, such as a product list in a shopping page or a price table in a hotel page. Some pages are organized with more rigorous structures, such as the home pages of the professors in a university. Other pages have very strict structures, such as the book description pages of Amazon, which are usually generated by a uniform template.

Therefore, basically two kinds of Web IE techniques exist: IE from unstructured pages and IE from semistructured pages. IE tools for unstructured pages are similar to those classical IE tools that typically use natural language processing techniques such as syntactic analysis, semantic analysis, and discourse analysis. IE tools for semistructured pages are different from the classical ones, as IE utilizes available structured information, such as HTML tags and page layouts, to infer the data formats or pages. Such a kind of methods is also called *wrapper induction* (Kushmerick, Weld, & Doorenbos, 1997; Cohen, Hurst, & Jensen, 2002). In contrast to classic IE approaches, wrapper induction operates less dependently of the specific contents of Web pages and mainly focuses on page structure and layout. Existing approaches for Web IE mainly include the manual approach, supervised learning, and unsupervised learning. Although some manually built wrappers exist, supervised learning and unsupervised learning are viewed as more promising ways to learn robust and scalable wrappers, because building IE tools manually is not feasible and scalable for the dynamic, massive and diverse Web contents. Moreover, because supervised learning still relies on manually labeled sample pages and thus also requires substantial human effort, unsupervised learning is the most suitable method for Web IE. There have been several successful fully automatic IE tools using unsupervised learning (Arasu & Garcia-Molina, 2003; Liu, Grossman, & Zhai, 2003).

Deep Web Mining

In the deep Web, it is usually difficult or even impossible to directly obtain the structures (i.e. schemas) of the Web sites' backend databases without cooperation from the sites. Instead, the sites present two other distinguishing structures, interface schema and result schema, to users. The *interface schema* is the schema of the query interface, which exposes attributes that can be queried in the backend database. The *result schema* is the schema of the query results, which exposes attributes that are shown to users.

The interface schema is useful for applications, such as a mediator that queries multiple Web databases, because the mediator needs complete knowledge about the search interface of each database. The result schema is critical for applications, such as data extraction, where instances in the query results are extracted. In addition to the importance of the interface schema and result schema, attribute matching across different schemas is also important. First, matching between different interface schemas and matching between different results schemas (intersite schema matching) are critical for metasearching and data-integration among related Web databases. Second, matching between the interface schema and the result schema of a single Web database (intrasite schema matching) enables automatic data annotation and database content crawling.

Most existing schema-matching approaches for Web databases primarily focus on matching query interfaces (He & Chang, 2003; He, Meng, Yu, & Wu, 2003; Raghavan & Garcia-Molina, 2001). They usually adopt a label-based strategy to identify attribute labels from the descriptive text surrounding interface elements and then find synonymous relationships between the identified labels. The performance of these approaches may be affected when no attribute description can be identified or when the identified description is not informative. In Wang, Wen, Lochovsky, and Ma (2004), an instance-based schema-matching approach was proposed to identify both the interface and result schemas of Web databases. Instance-based approaches depend on the content overlap or statistical properties, such as data ranges and patterns, to determine the similarity of two attributes. Thus, they could effectively deal with the cases where attribute names or labels are missing or not available, which are common for Web databases.

FUTURE TRENDS

It is foreseen that the biggest challenge in the next several decades is how to effectively and efficiently dig out a machine-understandable information and knowledge layer from unorganized and unstructured Web data. However, Web structure mining techniques are still in their youth today. For example, the accuracy of Web information extraction tools, especially those automatically learned tools, is still not satisfactory to meet the requirements of some rigid applications. Also, deep Web mining is a new area, and researchers have many challenges and opportunities to further explore, such as data extraction, data integration, schema learning and matching, and so forth. Moreover, besides Web pages, various other types of structured data exist on the Web, such as e-mail, newsgroup, blog, wiki, and so forth. Applying Web mining techniques to extract structures from these data types is also a very important future research direction.

CONCLUSION

Despite the efforts of XML and semantic Web, which target to bring structures and semantics to the Web, Web structure mining is considered a more promising way to structuralize the Web due to its characteristics of automation, scalability, generality, and robustness. As a result, there has been a rapid growth of technologies for automatically discovering structures from the Web, namely Web graph mining, Web information extraction, and deep Web mining. The mined information and knowledge will greatly improve the effectiveness of current Web search and will enable much more sophisticated Web information retrieval technologies in the future.

REFERENCES

Arasu, A., & Garcia-Molina, H. (2003). Extracting structured data from Web pages. *Proceedings of the ACM SIGMOD International Conference on Management of Data*.

Bharat, K., & Henzinger, M. R. (1998). Improved algorithms for topic distillation in a hyperlinked envi-

ronment. *Proceedings of the 21st Annual International ACM SIGIR Conference*.

Cai, D., He, X., Wen J.-R., & Ma, W.-Y. (2004). Block-level link analysis. *Proceedings of the 27th Annual International ACM SIGIR Conference*.

Chakrabarti, S., Joshi, M., & Tawde, V. (2001). Enhanced topic distillation using text, markup tags, and hyperlinks. *Proceedings of the 24th Annual International ACM SIGIR Conference* (pp. 208-216).

Chang, C. H., He, B., Li, C., & Zhang, Z. (2003). *Structured databases on the Web: Observations and implications discovery* (Tech. Rep. No. UIUCCDCS-R-2003-2321). Urbana-Champaign, IL: University of Illinois, Department of Computer Science.

Cohen, W., Hurst, M., & Jensen, L. (2002). A flexible learning system for wrapping tables and lists in HTML documents. *Proceedings of the 11th World Wide Web Conference*.

Flake, G. W., Lawrence, S., & Lee Giles, C. (2000). Efficient identification of Web communities. *Proceedings of the Sixth International Conference on Knowledge Discovery and Data Mining*.

Florescu, D., Levy, A. Y., & Mendelzon, A. O. (1998). Database techniques for the World Wide Web: A survey. *SIGMOD Record, 27*(3), 59-74.

Gibson, D., Kleinberg, J., & Raghavan, P. (1998). Inferring Web communities from link topology. *Proceedings of the Ninth ACM Conference on Hypertext and Hypermedia*.

He, B., & Chang, C. C. (2003). Statistical schema matching across Web query interfaces. *Proceedings of the ACM SIGMOD International Conference on Management of Data*.

He, H., Meng, W., Yu, C., & Wu, Z. (2003). WISE-Integrator: An automatic integrator of Web search interfaces for e-commerce. *Proceedings of the 29th International Conference on Very Large Data Bases*.

Kleinberg, J. (1998). Authoritative sources in a hyperlinked environment. *Proceedings of the Ninth ACM SIAM International Symposium on Discrete Algorithms* (pp. 668-677).

Kumar, R., Raghavan, P., Rajagopalan, S., & Tomkins, A. (1999). Trawling the Web for emerging cyber-communities. *Proceedings of the Eighth International World Wide Web Conference*.

Kushmerick, N., Weld, D., & Doorenbos, R. (1997). Wrapper induction for information extraction. *Proceedings of the International Joint Conference on Artificial Intelligence*.

Liu, B., Grossman, R., & Zhai, Y. (2003). Mining data records in Web pages. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

Page, L., Brin, S., Motwani, R., & Winograd, T. (1998). *The PageRank citation ranking: Bringing order to the Web* (Tech. Rep.). Stanford University.

Popescul, A., Flake, G. W., Lawrence, S., Ungar, L. H., & Lee Giles, C. (2000). Clustering and identifying temporal trends in document databases. *Proceedings of the IEEE Conference on Advances in Digital Libraries*.

Raghavan, S., & Garcia-Molina, H. (2001). Crawling the hidden Web. *Proceedings of the 27th International Conference on Very Large Data Bases*.

Wang, J., Wen, J.-R., Lochovsky, F., & Ma, W.-Y. (2004). Instance-based schema matching for Web databases by domain-specific query probing. *Proceedings of the 30th International Conference on Very Large Data Bases*.

KEY TERMS

Community Mining: A Web graph mining algorithm to discover communities from the Web graph in order to provide a higher logical view and more precise insight of the nature of the Web.

Deep Web Mining: Automatically discovering the structures of Web databases hidden in the deep Web and matching semantically related attributes between them.

HITS: A Web graph mining algorithm to compute authority scores and hub scores for Web pages.

PageRank: A Web graph mining algorithm that uses the probability that a page is visited by a random surfer on the Web as a key factor for ranking search results.

Web Graph Mining: The mining techniques used to discover knowledge from the Web graph.

Web Information Extraction: The class of mining methods to pull out information from a collection of Web pages and converting it to a homogeneous form that is more readily digested and analyzed for both humans and machines.

Web Structure Mining: The class of methods used to automatically discover structured data and information from the Web.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 443-447, copyright 2005 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

Ensemble Data Mining Methods

Nikunj C. Oza
NASA Ames Research Center, USA

INTRODUCTION

Ensemble Data Mining Methods, also known as Committee Methods or Model Combiners, are machine learning methods that leverage the power of multiple models to achieve better prediction accuracy than any of the individual models could on their own. The basic goal when designing an ensemble is the same as when establishing a committee of people: each member of the committee should be as competent as possible, but the members should be complementary to one another. If the members are not complementary, that is, if they always agree, then the committee is unnecessary—any one member is sufficient. If the members are complementary, then when one or a few members make an error, the probability is high that the remaining members can correct this error. Research in ensemble methods has largely revolved around designing ensembles consisting of competent yet complementary models.

BACKGROUND

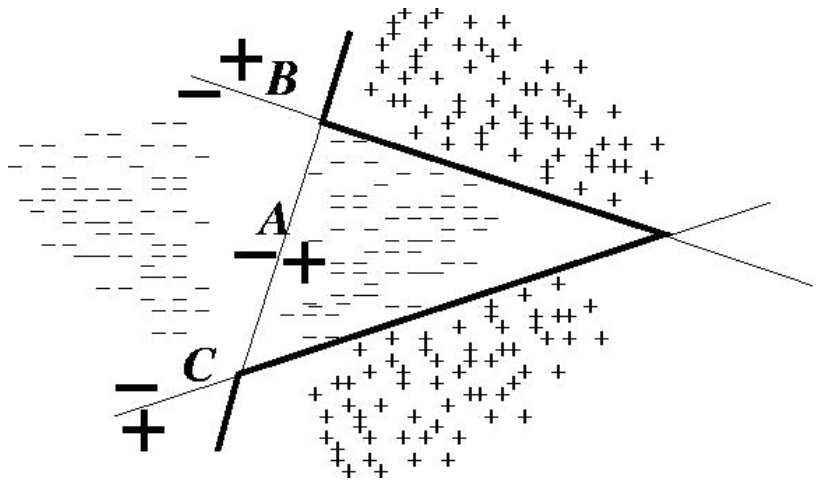
A supervised machine learner constructs a mapping from input data (normally described by several features) to the appropriate outputs. It does this by learning from a training set—N inputs $x_1, x_2, \ldots, x_N$ for which the corresponding true outputs $y_1, y_2, \ldots, y_N$ are known. The model that results is used to map new inputs to the appropriate outputs. In a classification learning task, each output is one or more classes to which the input belongs. The goal of classification learning is to develop a model that separates the data into the different classes, with the aim of classifying new examples in the future. For example, a credit card company may develop a model that separates people who defaulted on their credit cards from those who did not based on other known information such as annual income. A model would be generated based on data from past credit card holders. The model would be used to predict whether a new credit card applicant is likely to default on his credit card and thereby decide whether to approve or deny this applicant a new card. In a regression learning task, each output is a continuous value to be predicted (e.g., the average balance that a credit card holder carries over to the next month).

Many traditional machine learning algorithms generate a single model (e.g., a decision tree or neural network). Ensemble learning methods instead generate multiple models. Given a new example, the ensemble passes it to each of its multiple *base* models, obtains their predictions, and then combines them in some appropriate manner (e.g., averaging or voting). As mentioned earlier, it is important to have base models that are competent but also complementary (Tumer and Ghosh, 1996). To further motivate this point, consider Figure 1. This figure depicts a classification problem in which the goal is to separate the points marked with plus signs from points marked with minus signs. None of the three individual linear classifiers (marked A, B, and C) is able to separate the two classes of points. However, a majority vote over all three linear classifiers yields the piecewise-linear classifier shown as a thick line. This classifier is able to separate the two classes perfectly. For example, the plusses at the top of the figure are correctly classified by A and B, but are misclassified by C. The majority vote over these correctly classifies these points as plusses. This happens because A and B are very different from C. If our ensemble instead consisted of three copies of C, then all three classifiers would misclassify the plusses at the top of the figure, and so would a majority vote over these classifiers.

MAIN THRUST OF THE CHAPTER

We now discuss the key elements of an ensemble learning method and ensemble model and, in the process, discuss several ensemble methods that have been developed.

Figure 1. An ensemble of linear classifiers. Each line—A, B, and C—is a linear classifier. The boldface line is the ensemble that classifies new examples by returning the majority vote of A, B, and C



Ensemble Methods

The example shown in Figure 1 is an artificial example. We cannot normally expect to obtain base models that misclassify examples in completely separate parts of the input space and ensembles that classify all the examples correctly. However, there are many algorithms that attempt to generate a set of base models that make errors that are as different from one another as possible. Methods such as Bagging (Breiman, 1994) and Boosting (Freund and Schapire, 1996) promote diversity by presenting each base model with a different subset of training examples or different weight distributions over the examples. For example, in figure 1, if the plusses in the top part of the figure were temporarily removed from the training set, then a linear classifier learning algorithm trained on the remaining examples would probably yield a classifier similar to C. On the other hand, removing the plusses in the bottom part of the figure would probably yield classifier B or something similar. In this way, running the same learning algorithm on different subsets of training examples can yield very different classifiers which can be combined to yield an effective ensemble. Input Decimation Ensembles (IDE) (Tumer and Oza, 2003) and Stochastic Attribute Selection Committees (SASC) (Zheng and Webb, 1998) instead promote diversity by training each base model with the same training examples but different subsets of the input features. SASC trains each base model with a random subset of input features. IDE selects, for each class, a subset of features that has the highest correlation with the presence or absence of that class. Each feature subset is used to train one base model. However, in both SASC and IDE, all the training patterns are used with equal weight to train all the base models.

So far we have distinguished ensemble methods by the way they train their base models. We can also distinguish methods by the way they combine their base models' predictions. Majority or plurality voting is frequently used for classification problems and is used in Bagging. If the classifiers provide probability values, simple averaging is commonly used and is very effective (Tumer and Ghosh, 1996). Weighted averaging has also been used and different methods for weighting the base models have been examined. Two particularly interesting methods for weighted averaging include Mixtures of Experts (Jordan and Jacobs, 1994) and Merz's use of Principal Components Analysis (PCA) to combine models (Merz, 1999). In Mixtures of Experts, the weights in the weighted average combining are determined by a gating network, which is a model that takes the same inputs that the base models take, and returns a weight for each of the base models. The higher the weight for a base model, the more that base model is trusted to provide the correct answer. These weights are determined during training by how well the base models perform on the training examples. The gating network essentially keeps track of how well each base model performs in each part of the input

space. The hope is that each model learns to specialize in different input regimes and is weighted highly when the input falls into its specialty. Intuitively, we regularly use this notion of giving higher weights to opinions of experts with the most appropriate specialty. Merz's method uses PCA to lower the weights of base models that perform well overall but are redundant and therefore effectively give too much weight to one model. For example, in Figure 1, if there were instead two copies of A and one copy of B in an ensemble of three models, we may prefer to lower the weights of the two copies of A since, essentially, A is being given too much weight. Here, the two copies of A would always outvote B, thereby rendering B useless. Merz's method also increases the weight on base models that do not perform as well overall but perform well in parts of the input space where the other models perform poorly. In this way, a base model's unique contributions are rewarded. Many of the methods described above have been shown to be specific cases of one method: Importance Sampled Learning Ensembles (Friedman and Popescu, 2003).

When designing an ensemble learning method, in addition to choosing the method by which to bring about diversity in the base models and choosing the combining method, one has to choose the type of base model and base model learning algorithm to use. The combining method may restrict the types of base models that can be used. For example, to use average combining in a classification problem, one must have base models that can yield probability estimates. This precludes the use of linear discriminant analysis which is not set up to return probabilities. The vast majority of ensemble methods use only one base model learning algorithm but use the methods described earlier to bring about diversity in the base models. There has been surprisingly little work (e.g., (Merz 1999)) on creating ensembles with many different types of base models.

Two of the most popular ensemble learning algorithms are Bagging and Boosting, which we briefly explain next.

Bagging

*B*ootstrap *Agg*regat*ing* (Bagging) generates multiple bootstrap training sets from the original training set (using sampling with replacement) and uses each of them to generate a classifier for inclusion in the ensemble. The algorithms for bagging and sampling with replacement are given in Figure 2. In these algorithms, $\{(x_1,y_1), (x_2,y_2), \ldots, (x_N,y_N)\}$ is the training set., M is the number of base models to be learned, L_b is the base model learning algorithm, the h_i's are the base models, *random_integer(a,b)* is a function that returns each of the integers from a to b with equal probability, and $I(X)$ is the indicator function that returns 1 if X is true and 0 otherwise.

To create a bootstrap training set from an original training set of size N, we perform N Multinomial trials, where in each trial, we draw one of the N examples. Each example has probability $1/N$ of being drawn in each trial. The second algorithm shown in figure 2 does

Figure 2. Batch bagging algorithm and sampling with replacement

Bagging($\{(x_1, y_1), (x_2, y_2),\ldots,(x_N, y_N)\}$,$M$)
For each m = 1,2,...,M
T_m = Sample_With_Replacement($\{(x_1, y_1), (x_2, y_2),\ldots,(x_N, y_N)\}$,$N$)
$h_m = L_b(T_m)$
Return $h_{fin}(x) = \text{argmax}_{y \in Y} \sum_{m=1}^{M} I(h_m(x) = y)$.

Sample_With_Replacement (T,N)
S = {}
For i = 1,2,...,N
    r = random_integer(1,N)
    Add $T[r]$ to S.
Return S.

exactly this—N times, the algorithm chooses a number r from 1 to N and adds the rth training example to the bootstrap training set S. Clearly, some of the original training examples will not be selected for inclusion in the bootstrap training set and others will be chosen one time or more. In bagging, we create M such bootstrap training sets and then generate classifiers using each of them. Bagging returns a function $h(x)$ that classifies new examples by returning the class y that gets the maximum number of votes from the base models $h_1,h_2,\ldots,h_M$. In bagging, the M bootstrap training sets that are created are likely to have some differences. (Breiman, 1994) demonstrates that bagged ensembles tend to improve upon their base models more if the base model learning algorithms are *unstable*—differences in their training sets tend to induce significant differences in the models. He notes that decision trees are unstable, which explains why bagged decision trees often outperform individual decision trees; however, decision stumps (decision trees with only one variable test) are stable, which explains why bagging with decision stumps tends not to improve upon individual decision stumps.

Boosting

Boosting algorithms are a class of algorithms that have been mathematically proven to improve upon the performance of their base models in certain situations. We now explain the AdaBoost algorithm because it is the most frequently used among all boosting algorithms. AdaBoost generates a sequence of base models with different weight distributions over the training set. The AdaBoost algorithm is shown in Figure 3. Its inputs are a set of N training examples, a base model learning algorithm L_b, and the number M of base models that we wish to combine. AdaBoost was originally designed for two-class classification problems; therefore, for this explanation we will assume that there are two possible classes. However, AdaBoost is regularly used with a larger number of classes. The first step in AdaBoost is to construct an initial distribution of weights D_1 over the training set. This distribution assigns equal weight to all N training examples. We now enter the loop in the algorithm. To construct the first base model, we call L_b with distribution D_1 over the training set. After getting back a model h_1, we calculate its error ε_1 on the training set itself, which is just the sum of the weights of the training examples that h_1 misclassifies. We require that $\varepsilon_1 < 1/2$ (this is the *weak learning* assumption—the error should be less than what we would achieve through randomly guessing the class).[1] If this condition is not satisfied, then we stop and return the ensemble consisting of the previously-generated base models. If this condition is satisfied, then we calculate a new distribution D_2 over the training examples as

Figure 3. AdaBoost algorithm

AdaBoost($\{(x_1, y_1), (x_2, y_2),\ldots,(x_N, y_N)\}$,$L_b$,$M$)

Initialize $D_1(n) = 1/N$ for all $n \in \{1,2,\ldots,N\}$.

For each $m = 1,2,\ldots,M$:

$h_m = L_b(\{(x_1, y_1), (x_2, y_2),\ldots,(x_N, y_N)\},D_m)$

$\varepsilon_m = \sum_{n:h_m(x_n)\neq y_n} D_m(n)$.

If $\varepsilon_m \geq 1/2$ then,

set $M = m - 1$ and abort this loop.

Update distribution D_m:

$$D_{m+1}(n) = D_m(n) \times \begin{cases} \dfrac{1}{2(1-\varepsilon_m)} & \text{if } h_m(x_n) = y_n \\ \dfrac{1}{2\varepsilon_m} & \text{otherwise.} \end{cases}$$

Return $h_{fin}(x) = \text{argmax}_{y\in Y} \sum_{m=1}^{M} I(h_m(x) = y) \log \dfrac{1-\varepsilon_m}{\varepsilon_m}$.

follows. Examples that were correctly classified by h_1 have their weights multiplied by $1/(2(1-\varepsilon_1))$. Examples that were misclassified by h_1 have their weights multiplied by $1/(2\varepsilon_1)$. Note that, because of our condition $\varepsilon_1 < 1/2$, correctly classified examples have their weights reduced and misclassified examples have their weights increased. Specifically, examples that h_1 misclassified have their total weight increased to $1/2$ under D_2 and examples that h_1 correctly classified have their total weight reduced to $1/2$ under D_2. We then go into the next iteration of the loop to construct base model h_2 using the training set and the new distribution D_2. The point is that the next base model will be generated by a weak learner (i.e., the base model will have error less than $1/2$); therefore, at least some of the examples misclassified by the previous base model will have to be correctly classified by the current base model. In this way, boosting forces subsequent base models to correct the mistakes made by earlier models. We construct M base models in this fashion. The ensemble returned by AdaBoost is a function that takes a new example as input and returns the class that gets the maximum weighted vote over the M base models, where each base model's weight is $\log((1-\varepsilon_m)/\varepsilon_m)$, which is proportional to the base model's accuracy on the weighted training set presented to it.

AdaBoost has performed very well in practice and is one of the few theoretically-motivated algorithms that has turned into a practical algorithm. However, AdaBoost can perform poorly when the training data is noisy (Dietterich, 2000), i.e., the inputs or outputs have been randomly contaminated. Noisy examples are normally difficult to learn. Because of this, the weights assigned to noisy examples often become much higher than for the other examples, often causing boosting to focus too much on those noisy examples at the expense of the remaining data. Some work has been done to mitigate the effect of noisy examples on boosting (Oza 2004, Ratsch et. al., 2001).

FUTURE TRENDS

The fields of machine learning and data mining are increasingly moving away from working on small datasets in the form of flat files that are presumed to describe a single process. The fields are changing their focus toward the types of data increasingly being encountered today: very large datasets, possibly distributed among different locations, describing operations with multiple modes, time-series data, online applications (the data is not a time series but nevertheless arrives continually and must be processed as it arrives), partially-labeled data, and documents. Research in ensemble methods is beginning to explore these new types of data. For example, ensemble learning traditionally has required access to the entire dataset at once, i.e., it performs batch learning. However, this is clearly impractical for very large datasets that cannot be loaded into memory all at once. Some recent work (Oza and Russell, 2001; Oza, 2001) applies ensemble learning to such large datasets. In particular, this work develops online versions of bagging and boosting. That is, whereas standard bagging and boosting require at least one scan of the dataset for every base model created, online bagging and online boosting require only one scan of the dataset regardless of the number of base models. Additionally, as new data arrives, the ensembles can be updated without reviewing any past data. However, because of their limited access to the data, these online algorithms do not perform as well as their standard counterparts. Other work has also been done to apply ensemble methods to other types of problems such as remote sensing (Rajan and Ghosh, 2005), person recognition (Chawla and Bowyer, 2005), one vs. all recognition (Cabrera et. al., 2005), and medicine (Pranckeviciene et. al., 2005)—a recent survey of such applications is (Oza and Tumer, 2008). However, most of this work is experimental. Theoretical frameworks that can guide us in the development of new ensemble learning algorithms specifically for modern datasets have yet to be developed.

CONCLUSION

Ensemble Methods began about fifteen years ago as a separate research area within machine learning and were motivated by the idea of wanting to leverage the power of multiple models and not just trust one model built on a small training set. Significant theoretical and experimental developments have occurred over the past fifteen years and have led to several methods, especially bagging and boosting, being used to solve many real problems. However, ensemble methods also appear to be applicable to current and upcoming problems of distributed data mining, online applications, and others. Therefore, practitioners in data mining should stay tuned for further developments in the vibrant area

of ensemble methods. An excellent way to do this is to read a recent textbook on ensembles (Kuncheva, 2004) and follow the series of workshops called the International Workshop on Multiple Classifier Systems (proceedings published by Springer). This series' balance between theory, algorithms, and applications of ensemble methods gives a comprehensive idea of the work being done in the field.

REFERENCES

Breiman, L. (1994). *Bagging predictors*. Technical Report 421, Department of Statistics, University of California, Berkeley.

Cabrera, J.B.D., Gutierrez, C., & Mehra, R.K. (2005). Infrastructures and Algorithms for Distributed Anomaly-based Intrusion Detection in Mobile Ad-hoc Networks. In *Proceedings of the IEEE Conference on Military Communications*, pp. 1831-1837. IEEE, Atlantic City, New Jersey, USA.

Chawla, N., & Bowyer, K. (2005). Designing multiple classifier systems for face recognition. In N. Oza, R. Polikar, J. Kittler, & F. Roli, (Eds.), *Proceedings of the Sixth International Workshop on Multiple Classifier Systems*, pp. 407-416. Springer-Verlag.

Dietterich, T. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning, 40,* 139-158.

Freund, Y., & Schapire, R. (1996). Experiments with a new boosting algorithm. In *Proceedings of the Thirteenth International Conference on Machine Learning*, pp. 148-156. Morgan Kaufmann.

Friedman, J.H., & Popescu, B.E. (2003). *Importance sampled learning ensembles*. Technical Report, Stanford University.

Jordan, M.I., & Jacobs, R.A. (1994). Hierarchical mixture of experts and the EM algorithm. *Neural Computation, 6,* 181-214.

Kuncheva, L.I. (2004). *Combining pattern classifiers: Methods and algorithms*. Wiley-Interscience.

Merz, C.J. (1999). A principal component approach to combining regression estimates. *Machine Learning, 36*, 9-32.

Oza, N.C. (2001). *Online ensemble learning*. PhD thesis, University of California, Berkeley.

Oza, N.C. (2004). AveBoost2: Boosting with noisy data. In F. Roli, J. Kittler, & T. Windeatt (Eds.), *Proceedings of the Fifth International Workshop on Multiple Classifier Systems*, pp. 31-40, Springer-Verlag.

Oza, N.C. & Russell, S. (2001). Experimental comparisons of online and batch versions of bagging and boosting. *The Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, California.

Oza, N.C. & Tumer, K. (2008). Classifier ensembles: Select real-world applications. *Information Fusion, Special Issue on Applications of Ensemble Methods, 9*(1), 4-20.

Prackeviciene, E., Baumgartner, R., & Somorjai, R. (2005). Using domain knowledge in the random subspace method: Application to the classification of biomedical spectra. In N. C. Oza, R. Polikar, J, Kittler, & F, Roli (Eds.), *Proceedings of the Sixth International Workshop on Multiple Classifier Systems*, pp. 336-345. Springer-Verlag.

Rajan, S., & Ghosh, J. (2005). Exploiting class hierarchies for knowledge transfer in hyperspectral data. In N.C. Oza, R. Polikar, J. Kittler, & F. Roli (Eds.), *Proceedings of the Sixth International Workshop on Multiple Classifier Systems*, pp. 417-428. Springer-Verlag.

Ratsch, G., Onoda, T., & Muller, K.R. (2001). Soft margins for AdaBoost. *Machine Learning, 42,* 287-320.

Tumer, K. & Ghosh, J. (1996). Error correlation and error reduction in ensemble classifiers. *Connection Science, Special Issue on Combining Artificial Neural Networks: Ensemble Approaches, 8*(3 & 4), 385-404.

Tumer, K. & Oza, N.C. (2003). Input decimated ensembles. *Pattern Analysis and Applications, 6*(1), 65-77.

Zheng, Z. & Webb, G. (1998). Stochastic attribute selection committees. In *Proceedings of the 11th Australian Joint Conference on Artificial Intelligence (AI'98),* pp. 321-332.

KEY TERMS

Batch Learning: Learning using an algorithm that views the entire dataset at once and can access any part of the dataset at any time and as many times as desired.

Decision Tree: A model consisting of nodes that contain tests on a single attribute and branches representing the different outcomes of the test. A prediction is generated for a new example by performing the test described at the root node and then proceeding along the branch that corresponds to the outcome of the test. If the branch ends in a prediction, then that prediction is returned. If the branch ends in a node, then the test at that node is performed and the appropriate branch selected. This continues until a prediction is found and returned.

Ensemble: A function that returns a combination of the predictions of multiple machine learning models.

Machine Learning: The branch of Artificial Intelligence devoted to enabling computers to learn.

Neural Network: A nonlinear model derived through analogy with the human brain. It consists of a collection of elements that linearly combine their inputs and pass the result through a nonlinear transfer function.

Online Learning: Learning using an algorithm that only examines the dataset once in order. This paradigm is often used in situations when data arrives continually in a stream and predictions must be obtainable at any time.

Principal Components Analysis (PCA): Given a dataset, PCA determines the axes of maximum variance. For example, if the dataset was shaped like an egg, then the long axis of the egg would be the first principal component because the variance is greatest in this direction. All subsequent principal components are found to be orthogonal to all previous components.

ENDNOTE

[1] This requirement is perhaps too strict when there are more than two classes. There is a multi-class version of AdaBoost (Freund and Schapire, 1997) that does not have this requirement. However, the AdaBoost algorithm presented here is often used even when there are more than two classes if the base model learning algorithm is strong enough to satisfy the requirement.

Ensemble Learning for Regression

Niall Rooney
University of Ulster, UK

David Patterson
University of Ulster, UK

Chris Nugent
University of Ulster, UK

INTRODUCTION

The concept of ensemble learning has its origins in research from the late 1980s/early 1990s into combining a number of artificial neural networks (ANNs) models for regression tasks. Ensemble learning is now a widely deployed and researched topic within the area of machine learning and data mining. Ensemble learning, as a general definition, refers to the concept of being able to apply more than one learning model to a particular machine learning problem using some method of integration. The desired goal of course is that the ensemble as a unit will outperform any of its individual members for the given learning task. Ensemble learning has been extended to cover other learning tasks such as classification (refer to Kuncheva, 2004 for a detailed overview of this area), online learning (Fern & Givan, 2003) and clustering (Strehl & Ghosh, 2003). The focus of this article is to review ensemble learning with respect to regression, where by regression, we refer to the supervised learning task of creating a model that relates a continuous output variable to a vector of input variables.

BACKGROUND

Ensemble learning consists of two issues that need to be addressed, *ensemble generation*: how does one generate the base models/members of the ensemble and how large should the ensemble size be and *ensemble integration:* how does one integrate the base models' predictions to improve performance? Some ensemble schemes address these issues separately, others such as Bagging (Breiman, 1996a) and Boosting (Freund & Schapire, 1996) do not. The problem of ensemble generation where each base learning model uses the same learning algorithm (*homogeneous learning*) is generally addressed by a number of different techniques: using different samples of the training data or feature subsets for each base model or alternatively, if the learning method has a set of learning parameters, these may be adjusted to have different values for each of the base models. An alternative generation approach is to build the models from a set of different learning algorithms (*heterogeneous learning*). There has been less research in this latter area due to the increased complexity of effectively combining models derived from different algorithms. Ensemble integration can be addressed by either one of two mechanisms: either the predictions of the base models are combined in some fashion during the application phase to give an ensemble prediction (*combination approach*) or the prediction of one base model is selected according to some criteria to form the final prediction (*selection approach*). Both selection and combination can be either *static* in approach, where the learned model does not alter, or *dynamic* in approach, where the prediction strategy is adjusted for each test instance.

Theoretical and empirical work has shown that if an ensemble technique is to be effective, it is important that the base learning models are sufficiently *accurate* and *diverse* in their predictions (Hansen & Salomon, 1990; Sharkey, 1999; Dietterich, 2000). For regression problems, accuracy is usually measured based on the training error of the ensemble members and diversity is measured based on the concept of ambiguity or variance of the ensemble members' predictions (Krogh & Vedelsby, 1995). A well known technique to analyze the nature of supervised learning methods is based on the bias-variance decomposition of the expected error for a given target instance (Geman et al., 1992). In effect, the expected error can be represented by three terms, the

irreducible or random error, the bias (or squared bias) and the variance. The irreducible error is independent of the learning algorithm and places an upper bound on the performance of any regression technique. The bias term is a measure of how closely the learning algorithm's mean prediction over all training sets of fixed size, is near to the target. The variance is a measure of how the learning algorithms predictions for a given target, vary around the mean prediction. The purpose of an ensemble is to try to reduce bias and/or variance in the error. For a linear combination of 1,..,*N* base models where each i^{th} base model's contribution to the ensemble prediction is weighted by a coefficient α_i and $\sum_{i=1..N} \alpha_i = 1$, Krogh & Vedelsby (1995) showed that the generalization error of the ensemble trained on a single data set can also be decomposed into two terms. The first term consists of the weighted error of the individual ensemble members (their *weighted error or average accuracy*) and the second term represents the variability of the ensemble members predictions referred to as the *ambiguity* (or diversity) of the ensemble. They demonstrated that as the ambiguity increases and the first term remains the same, the error of the ensemble decreases. Brown et al. (2005) extended this analysis by looking at the bias-variance decomposition of a similarly composed ensemble and determined that the expected generalization error of the ensemble (if each model was equally weighted) can be decomposed into the expected average individual error and expected ambiguity and showed that these terms are not completely independent. They showed that increasing ambiguity will lead to a reduction in the error variance of the ensemble, but it can also lead to an increase in the level of averaged error in the ensemble members. So, in effect, there is a trade off between the ambiguity/diversity of an ensemble and the accuracy of its members.

MAIN FOCUS

In this section we consider in detail ensemble generation and integration methods.

Ensemble Generation

Ensembles can be generated to increase the level of diversity in homogeneous base models using the following methods:

Vary the learning parameters: If the learning algorithm has learning parameters, set each base model to have different parameter values e.g. in the area of neural networks one can set each base model to have different initial random weights or a different topology (Sharkey, 1999) or in the case of regression trees, each model can be built using a different splitting criteria or pruning strategy. In the technique of random forests (Breiman, 2001), regression trees are grown using a random selection of features for each splitting choice or in addition, randomizing the splitting criteria itself (Geurts et al., 2006).

Vary the data employed: Each base model is built using samples of the data. Resampling methods include cross-validation, boot-strapping, sampling with replacement (employed in Bagging (Breiman, 1996a), and adaptive sampling (employed in Boosting methods). If there are sufficient data, sampling can be replaced by using disjoint training sets for each base model.

Vary the features employed: In this approach, each model is built by a training algorithm, with a variable sub-set of the features in the data. This method was given prominence in the area of classification by Ho (Ho 1998a; Ho 1998b) and consists of building each model using training data consisting of input features in a *r*-dimensional random subspace subset from the original *p*-dimensional feature space. Tsymbal et al. (2003) proposed a variant of this approach to allow variable length feature sub-sets.

Randomised Outputs: In this approach, rather than present different regressors with different samples of input data, each regressor is presented with the same training data, but with output values for each instance perturbed by a randomization process (Breiman, 2000).

Ensemble Integration

The integration of ensembles works by either combining the base models outputs in some fashion or using selection methods to choose the "best" base model. Specific combination/selection methods which learns a meta-model as an integration technique, are described as meta-methods.

Much of the early research involving ensemble learning for regression focused on the combination of ANNs (Sharkey, 1999). However many of these methods can be applied directly to any ensemble of regression models, regardless of the base algorithm

employed. Typically each member of the ensemble was built using the same learning method. Researchers then investigated the effects of having members with different topologies, different initial weights or different training or samples of training data. The outputs from the base models were then combined using a linear function $\sum_{i=1..N} \alpha_i \hat{f}_i(x)$ where N is the size of the ensemble, and α_i is the weighting for model $\hat{f}_i(x)$.

The most simple ensemble method for regression is referred to as the Basic Ensemble Method (BEM) (Perrone & Cooper, 1993). BEM sets the weights α_i:

$$\alpha_i = \frac{1}{N}$$

in other words, the base members are combined using an unweighted averaging mechanism. This method does not take into account the performance of the base models, but as will be seen later with the similar method of Bagging, it can be effective. There exists numerical methods to give more "optimal" weights (values of α) to each base model. The generalised ensemble method (GEM) (Perrone & Cooper, 1993) calculates α based on a symmetric covariance matrix C of the individual models' errors. Another approach to calculate the optimal weights is to use a linear regression (LR) algorithm. However both GEM and LR techniques may suffer from a numerical problem known as the multi-collinear problem. This problem occurs where one or more models can be expressed as a linear combination of one or more of the other models. This can lead to a numerical problem as both techniques require taking the inverse of matrices, causing large rounding errors and as a consequence producing weights that are far from optimal. One approach to ameliorate multi-collinear problems is to used weight regularization, an example of this has been previously mentioned when the weights are constrained to sum to one. Principal components regression (PCR) was developed by Merz & Pazzani (1999) to overcome the multi-collinear problem.

The main idea in PCR is to transform the original learned models to a new set of models using Principal Components Analysis (PCA). The new models are a decomposition of the original models' predictions into N independent components. The most useful initial components are retained and the mapping is reversed to calculate the weights for the original learned models.

Another technique is Mixture of Experts where the weighting coefficients are considered *dynamic*. Such dynamic coefficients are not constants but are functions of the test instance x that is, $\sum_{i=1..N} \alpha_i(x) \hat{f}_i(x)$. This approach requires a simultaneous training step both for the dynamic coefficients and the base members themselves, and as such is a method exclusive to ANN research (Hansen, 1999). Negative-correlation learning is also an intriguing ANN ensemble approach to ensemble combination where the ensemble members are not trained independently of each other, but rather all the individual ensemble networks are trained simultaneously and interact using a combined correlation penalty term in their error functions. As such negative correlation learning tries to create networks which are negatively correlated in their errors (Liu & Yao, 1999; Brown & Wyatt, 2003), i.e. the errors between networks are inversely related. Granitto et al. (2005) use a simulated annealing approach to find the optimal number of training epochs for each ensemble member as determined by calculating the generalization error for a validation set.

Bagging (Bootstrap aggregation) (Breiman 1996a) averages the outputs of each based model where each base model is learned using different training sets generated by sampling with replacement. Breiman showed that this technique is a variance reducing one and is suited for unstable learners that typically have high variance in their error such as regression trees.

Leveraging techniques are based on the principle that the sequential combination of weak regressors based on a simple learning algorithm can be made to produce a strong ensemble of which boosting is a prime example (Duffy & Helmbold, 2002) The main purpose of boosting is to sequentially apply a regression algorithm to repeatedly modified versions of the data, thereby producing a sequence of regressors. The modified versions of the data are based on a sampling process related to the training error of individual instances in the training set. The most well known version of boosting is AdaBoost (adaptive boosting for regression) which combines regressors using the weighted median. Hastie et al. (2001) showed that AdaBoost is equivalent to a forward stage additive model. In a forward stage additive model an unspecified basis function is estimated for each predictor variable and added iteratively to the model. In the case of boosting, the basis functions are the individual ensemble models.

If combination methods rely on the ensemble members being diverse and accurate, global selection methods, where one member of the ensemble is cho-

sen to make all ensemble predictions, clearly do not. The classic simple, static selection method is *Cross Validation* (Schaffer, 1993), which chooses the base model with least estimated error via a cross-validation process. The use of cross-validation ensures that the estimates of error are less biased as they are calculated from different test data to the data on which models are built. The obvious drawback to this approach is that *N* models are built whereas only one is in effect used. To overcome this limitation, researchers (Merz, 1995, Rooney et al., 2004), have focused on localized selection of the base learners, where a base learner is chosen for a query instance based on its performance in the localized region of instance space. This is generally referred to as *a dynamic* ensemble combination technique.

The concept of meta-learning for ensembles is based on the concept of using a learning model (referred to as a meta-model) to combine or select from the base models. Note that this definition of meta-learning only refers to techniques described in this chapter. The most widely studied technique is known as Stacking. The concept of Stacking, (or in terms of regression, Stacked Regression(s)), was originally proposed by Wolpert (1992) and considered by LeBlanc and Tibshirani (1993) to be a bias reducing technique. However, its full range of theoretical properties is still not well understood.

The effectiveness of Stacking is based on its use of cross-validation during the training phase. As it builds the base regression models and estimates the weights using different samples of the training data, so as to reduce bias in the ensemble predictions. Combination is carried out using a learning algorithm or meta-model, rather than a simple linear combination. Originally a meta-model was proposed using a least squares linear regression function to determine the weights, Breiman (1996b) showed that for this to be effective for regression problems, the weights had to be constrained to be non-zero. Of course the meta-model may be any learning algorithm (Hastie et al., 2001). If the meta-model for example is a regression tree, the meta-model is no longer trying to find optimal weights at a global level, but at a piecewise localized level. Rooney & Patterson (2007) combined Stacking with the related technique of Dynamic Integration (Tsymbal et al., 2003), to create a more effective ensemble meta-learner for regression.

FUTURE TRENDS

Notwithstanding the development of new techniques and advances in the theory of ensembles, we identify the following future trends of research

- *Selective combination of ensemble members:* As noted, an effective ensemble requires that each member should be both diverse and accurate. One approach to enabling this is to use hill climbing techniques to selectively adjust the members of the ensemble to combine by calculating metrics based on accuracy or diversity (Opitz & Shavlik, 1996, Carney & Cunningham, 2000) or more simply to selectively reduce the set of ensemble members (Zhou et al., 2002) Potentially this can create a more competent ensemble but also a less complex one. We expect further work in this area particularly in trying to produce small ensembles of transparent and interpretable models such as regression trees.
- *Computationally Efficient methods:* If ensemble techniques are going to be applied to large-scale data mining methods, they need to be computationally efficient. A potential direction of research is the parallelization of ensemble methods – intrinsically such methods as Bagging are easily parallelizable.
- *Heterogeneous ensemble learning:* As discussed most ensemble method use the same learning algorithm to generate base models – there is scope for techniques that effectively combine more than one base learning algorithm.

CONCLUSION

Ensemble learning is an important method of deploying more than one learning model to give improved predictive accuracy for a given learning problem. We have described how regression based ensembles are able to reduce the bias and/or variance of the generalization error and reviewed the main techniques that have been developed for the generation and integration of regression based ensembles.

REFERENCES

Breiman, L. (1996). Bagging predictors. *Machine Learning, 24*, 123-140.

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5-32.

Breiman, L. (2000). Randomizing outputs to increase prediction accuracy. *Machine Learning, 40*(3), 229-242.

Breiman, L. (1996). Stacked regressions. *Machine Learning, 24*, 49-64.

Brown, G., Wyatt, J., Harris, R., & Yao, X. (2005). Diversity creation methods: A survey and categorization. *Information Fusion, 6*(1), 5-20.

Brown, G., & Wyatt, J. L. (2003). The use of the ambiguity decomposition in neural network ensemble learning methods. *20th International Conference on Machine Learning (ICML'03), Washington DC, USA.*

Brown, G., Wyatt, J. L., & Tino, P. (2005). Managing diversity in regression ensembles. *Journal of Machine Learning Research, 6*, 1621-1650.

Carney, J. G., & Cunningham, P. (2000). Tuning diversity in bagged ensembles. *International Journal of Neural Systems, 10*(4), 267-279.

Dietterich, T. G. (2000). Ensemble methods in machine learning. *1st International Workshop on Multiple Classifier Systems,, LNCS 1857* 1-15.

Duffy, N., & Helmbold, D. (2002). Boosting methods for regression. *Machine Learning, 47*(2), 153-200.

Fern, A., & Givan, R. (2003). Online ensemble learning: An empirical study. *Machine Learning, 53*(1), 71-109.

Freund, Y., & Schapire, R. E. (1996). Experiments with a new boosting algorithm. *Machine Learning: Proceedings of the Thirteenth International Conference, 148*, 156.

Geman, S., Bienenstock, E., & Doursat, R. (1992). Neural networks and the bias/variance dilemma. *Neural Computation, 4*(1), 1-58.

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning, 63*(1), 3-42.

Granitto, P. M., Verdes, P. F., & Ceccatto, H. A. (2005). Neural network ensembles: Evaluation of aggregation algorithms. *Artificial Intelligence, 163*(2), 139-162

Hansen, J. (1999). Combining predictors: Comparison of five meta machine learning methods. *Information Sciences, 119*(1), 91-105.

Hansen, L. K., & Salamon, P. (1990). Neural network ensembles. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 12*(10), 993-1001.

Hastie, T., Tibshirani, R., & Friedman, J. H. (2001). *The elements of statistical learning: Data mining, inference, and prediction*, Springer.

Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 20*(8), 832-844.

Ho, T. K. (1998). Nearest neighbors in random subspaces. *Proceedings of the Joint IAPR International Workshops on Advances in Pattern Recognition*, 640-648.

Krogh, A., & Vedelsby, J. (1995). Neural network ensembles, cross validation, and active learning. *Advances in Neural Information Processing Systems, 7*, 231–238.

Kuncheva, L. I. (2004). *Combining pattern classifiers: Methods and algorithms*, Wiley-Interscience.

Leblanc, M., & Tibshirani, R. (1996). Combining estimates in regression and classification. *Journal of the American Statistical Association, 91*(436).

Liu, Y., & Yao, X. (1999). Ensemble learning via negative correlation. *Neural Networks, 12*(10), 1399-1404.

Merz, C. J. (1995). Dynamical selection of learning algorithms. In Fisher, D. Lenz, H.J. (Ed.), *Learning from data: Artificial intelligence and statistics, 5*. New York: Springer Verlag.

Merz, C. J., & Pazzani, M. J. (1999). A principal components approach to combining regression estimates. *Machine Learning, 36*(1), 9-32.

Opitz, D. W., & Shavlik, J. W. (1996). Generating accurate and diverse members of a neural-network ensemble. *NIPS, 8*, 535–541.

Perrone, M. P., & Cooper, L. N. (1992). When networks disagree: Ensemble methods for hybrid neural networks. In R. J. Mammone (Ed.), *Artificial neural networks for speech and vision* (pp. 146-162), Chapman and Hall.

Rooney, N., Patterson, D., Anand, S., & Tsymbal, A. (2004). Dynamic integration of regression models. *Proceedings of the 5th International Multiple Classifier Systems Workshop, LNCS, 3077*, 164–173.

Rooney, N. & Patterson, D. (2007). A fusion of Stacking with Dynamic Integration. *20th International Joint Conference on Artificial Intelligence (IJCAI-07)*, pp. 2844-2849, 2007.

Schaffer, C. (1993). Selecting a classification method by cross-validation. *Machine Learning, 13*(1), 135-143.

Sharkey, A. J., & Sharkey, N. E. (1999). *Combining artificial neural nets: Ensemble and modular multi-net systems*, Springer-Verlag New York, Inc. Secaucus, NJ, USA.

Strehl, A., & Ghosh, J. (2003). Cluster ensembles - A knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research, 3*(3), 583-617.

Tsymbal, A., Puuronen, S., & Patterson, D. W. (2003). Ensemble feature selection with the simple bayesian classification. *Information Fusion, 4*(2), 87-100.

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks, 5*(2), 241-259.

Zhou, Z. H., Wu, J., & Tang, W. (2002). Ensembling neural networks: Many could be better than all. *Artificial Intelligence, 137*(1), 239-263.

KEY TERMS

Ensemble: A collection of models.

Ensemble Generation: A method by which an ensemble is generated so that each base model is different. Examples include sampling of data, randomization, different base learning algorithms or different learning algorithm parameterizations.

Ensemble Integration: A technique by which the predictions of base models are combined. Examples include unweighted/weighted averaging.

Ensemble Learning: A learning process involving an ensemble of base models.

Generalization Performance: A measure how well a supervised learning model performs on unseen instances (usually based on an error measure).

Regression/Continuous Prediction: The area of machine learning devoted to developing a predictive model that approximates the hidden function associating a continuous dependent or target or output variable with a set of independent or input variables. There are numerous algorithms with different learning hypothesis spaces for the process of regression.

Regressor: Regression model.

Supervised Learning: The process by which a learning model is produced using training data consisting of instances, each with a vector of inputs and known output values.

Ethics of Data Mining

Jack Cook
Rochester Institute of Technology, USA

INTRODUCTION

Decision makers thirst for answers to questions. As more data is gathered, more questions are posed: Which customers are most likely to respond positively to a marketing campaign, product price change or new product offering? How will the competition react? Which loan applicants are most likely or least likely to default? The ability to raise questions, even those that currently cannot be answered, is a characteristic of a good decision maker. Decision makers no longer have the luxury of making decisions based on gut feeling or intuition. Decisions must be supported by data; otherwise decision makers can expect to be questioned by stockholders, reporters, or attorneys in a court of law. Data mining can support and often direct decision makers in ways that are often counterintuitive. Although data mining can provide considerable insight, there is an "inherent risk that what might be inferred may be private or ethically sensitive" (Fule & Roddick, 2004, p. 159).

Extensively used in telecommunications, financial services, insurance, customer relationship management (CRM), retail, and utilities, data mining more recently has been used by educators, government officials, intelligence agencies, and law enforcement. It helps alleviate data overload by extracting value from volume. However, data analysis is not data mining. Query-driven data analysis, perhaps guided by an idea or hypothesis, that tries to deduce a pattern, verify a hypothesis, or generalize information in order to predict future behavior is not data mining (Edelstein, 2003). It may be a first step, but it is not data mining. Data mining is the process of discovering and interpreting meaningful, previously hidden patterns in the data. It is not a set of descriptive statistics. Description is not prediction. Furthermore, the focus of data mining is on the process, not a particular technique, used to make reasonably accurate predictions. It is iterative in nature and generically can be decomposed into the following steps: (1) data acquisition through translating, cleansing, and transforming data from numerous sources, (2) goal setting or hypotheses construction, (3) data mining, and (4) validating or interpreting results.

The process of generating rules through a mining operation becomes an ethical issue, when the results are used in decision-making processes that affect people or when mining customer data unwittingly compromises the privacy of those customers (Fule & Roddick, 2004). Data miners and decision makers must contemplate ethical issues before encountering one. Otherwise, they risk not identifying when a dilemma exists or making poor choices, since all aspects of the problem have not been identified.

BACKGROUND

Technology has moral properties, just as it has political properties (Brey 2000; Feenberg, 1999; Sclove, 1995; Winner, 1980). Winner (1980) argues that technological artifacts and systems function like laws, serving as frameworks for public order by constraining individuals' behaviors. Sclove (1995) argues that technologies possess the same kinds of structural effects as other elements of society, such as laws, dominant political and economic institutions, and systems of cultural beliefs. Data mining, being a technological artifact, is worthy of study from an ethical perspective due to its increasing importance in decision making, both in the private and public sectors. Computer systems often function less as background technologies and more as active constituents in shaping society (Brey, 2000). Data mining is no exception. Higher integration of data mining capabilities within applications ensures that this particular technological artifact will increasingly shape public and private policies.

Data miners and decision makers obviously are obligated to adhere to the law. But ethics are oftentimes more restrictive than what is called for by law. Ethics are standards of conduct that are agreed upon by cultures and organizations. Supreme Court Justice Potter Stewart defines the difference between ethics and laws as knowing the difference between what you

have a right to do (legally, that is) and what is right to do. Sadly, a number of IS professionals either lack an awareness of what their company actually does with data and data mining results or purposely come to the conclusion that it is not their concern. They are enablers in the sense that they solve management's problems. What management does with that data or results is not their concern.

Most laws do not explicitly address data mining, although court cases are being brought to stop certain data mining practices. A federal court ruled that using data mining tools to search Internet sites for competitive information may be a crime under certain circumstances (Scott, 2002). In *EF Cultural Travel BV vs. Explorica Inc.* (No. 01-2000 1st Cir. Dec. 17, 2001), the First Circuit Court of Appeals in Massachusetts held that Explorica, a tour operator for students, improperly obtained confidential information about how rival EF's Web site worked and used that information to write software that gleaned data about student tour prices from EF's Web site in order to undercut EF's prices (Scott, 2002). In this case, Explorica probably violated the federal Computer Fraud and Abuse Act (18 U.S.C. Sec. 1030). Hence, the source of the data is important when data mining.

Typically, with applied ethics, a morally controversial practice, such as how data mining impacts privacy, "is described and analyzed in descriptive terms, and finally moral principles and judgments are applied to it and moral deliberation takes place, resulting in a moral evaluation, and operationally, a set of policy recommendations" (Brey, 2000, p. 10). Applied ethics is adopted by most of the literature on computer ethics (Brey, 2000). Data mining may appear to be morally neutral, but appearances in this case are deceiving. This paper takes an applied perspective to the ethical dilemmas that arise from the application of data mining in specific circumstances as opposed to examining the technological artifacts (i.e., the specific software and how it generates inferences and predictions) used by data miners.

MAIN THRUST

Computer technology has redefined the boundary between public and private information, making much more information public. Privacy is the freedom granted to individuals to control their exposure to others. A customary distinction is between relational and informational privacy. Relational privacy is the control over one's person and one's personal environment, and concerns the freedom to be left alone without observation or interference by others. Informational privacy is one's control over personal information in the form of text, pictures, recordings, and so forth (Brey, 2000).

Technology cannot be separated from its uses. It is the ethical obligation of any information systems (IS) professional, through whatever means he or she finds out that the data that he or she has been asked to gather or mine is going to be used in an unethical way, to act in a socially and ethically responsible manner. This might mean nothing more than pointing out why such a use is unethical. In other cases, more extreme measures may be warranted. As data mining becomes more commonplace and as companies push for even greater profits and market share, ethical dilemmas will be increasingly encountered. Ten common blunders that a data miner may cause, resulting in potential ethical or possibly legal dilemmas, are (Skalak, 2001):

1. Selecting the wrong problem for data mining.
2. Ignoring what the sponsor thinks data mining is and what it can and cannot do.
3. Leaving insufficient time for data preparation.
4. Looking only at aggregated results, never at individual records.
5. Being nonchalant about keeping track of the mining procedure and results.
6. Ignoring suspicious findings in a haste to move on.
7. Running mining algorithms repeatedly without thinking hard enough about the next stages of the data analysis.
8. Believing everything you are told about the data.
9. Believing everything you are told about your own data mining analyses.
10. Measuring results differently from the way the sponsor will measure them.

These blunders are hidden ethical dilemmas faced by those who perform data mining. In the next subsections, sample ethical dilemmas raised with respect to the application of data mining results in the public sector are examined, followed briefly by those in the private sector.

Ethics of Data Mining in the Public Sector

Many times, the objective of data mining is to build a customer profile based on two types of data—factual (who the customer is) and transactional (what the customer does) (Adomavicius & Tuzhilin, 2001). Often, consumers object to transactional analysis. What follows are two examples; the first (identifying successful students) creates a profile based primarily on factual data, and the second (identifying criminals and terrorists) primarily on transactional.

Identifying Successful Students

Probably the most common and well-developed use of data mining is the attraction and retention of customers. At first, this sounds like an ethically neutral application. Why not apply the concept of students as customers to the academe? When students enter college, the transition from high school for many students is overwhelming, negatively impacting their academic performance. High school is a highly structured Monday-through-Friday schedule. College requires students to study at irregular hours that constantly change from week to week, depending on the workload at that particular point in the course. Course materials are covered at a faster pace; the duration of a single class period is longer; and subjects are often more difficult. Tackling the changes in a student's academic environment and living arrangement as well as developing new interpersonal relationships is daunting for students. Identifying students prone to difficulties and intervening early with support services could significantly improve student success and, ultimately, improve retention and graduation rates.

Consider the following scenario that realistically could arise at many institutions of higher education. Admissions at the institute has been charged with seeking applicants who are more likely to be successful (i.e., graduate from the institute within a five-year period). Someone suggests data mining existing student records to determine the profile of the most likely successful student applicant. With little more than this loose definition of success, a great deal of disparate data is gathered and eventually mined. The results indicate that the most likely successful applicant, based on factual data, is an Asian female whose family's household income is between $75,000 and $125,000 and who graduates in the top 25% of her high school class. Based on this result, admissions chooses to target market such high school students. Is there an ethical dilemma? What about diversity? What percentage of limited marketing funds should be allocated to this customer segment? This scenario highlights the importance of having well-defined goals before beginning the data mining process. The results would have been different if the goal were to find the most diverse student population that achieved a certain graduation rate after five years. In this case, the process was flawed fundamentally and ethically from the beginning.

Identifying Criminals and Terrorists

The key to the prevention, investigation, and prosecution of criminals and terrorists is information, often based on transactional data. Hence, government agencies increasingly desire to collect, analyze, and share information about citizens and aliens. However, according to Rep. Curt Weldon (R-PA), chairman of the House Subcommittee on Military Research and Development, there are 33 classified agency systems in the federal government, but none of them link their raw data together (Verton, 2002). As Steve Cooper, CIO of the Office of Homeland Security, said, "I haven't seen a federal agency yet whose charter includes collaboration with other federal agencies" (Verton, 2002, p. 5). Weldon lambasted the federal government for failing to act on critical data mining and integration proposals that had been authored before the terrorists' attacks on September 11, 2001 (Verton, 2002).

Data to be mined is obtained from a number of sources. Some of these are relatively new and unstructured in nature, such as help desk tickets, customer service complaints, and complex Web searches. In other circumstances, data miners must draw from a large number of sources. For example, the following databases represent some of those used by the U.S. Immigration and Naturalization Service (INS) to capture information on aliens (Verton, 2002).

- Employment Authorization Document System
- Marriage Fraud Amendment System
- Deportable Alien Control System
- Reengineered Naturalization Application Casework System
- Refugees, Asylum, and Parole System
- Integrated Card Production System
- Global Enrollment System

- Arrival Departure Information System
- Enforcement Case Tracking System
- Student and Schools System
- General Counsel Electronic Management System
- Student Exchange Visitor Information System
- Asylum Prescreening System
- Computer-Linked Application Information Management System (two versions)
- Non-Immigrant Information System

There are islands of excellence within the public sector. One such example is the U.S. Army's Land Information Warfare Activity (LIWA), which is credited with "having one of the most effective operations for mining publicly available information in the intelligence community" (Verton, 2002, p. 5).

Businesses have long used data mining. However, recently, governmental agencies have shown growing interest in using "data mining in national security initiatives" (Carlson, 2003, p. 28). Two government data mining projects, the latter renamed by the euphemism "factual data analysis," have been under scrutiny (Carlson, 2003) These projects are the U.S. Transportation Security Administration's (TSA) Computer Assisted Passenger Prescreening System II (CAPPS II) and the Defense Advanced Research Projects Agency's (DARPA) Total Information Awareness (TIA) research project (Gross, 2003). TSA's CAPPS II will analyze the name, address, phone number, and birth date of airline passengers in an effort to detect terrorists (Gross, 2003). James Loy, director of the TSA, stated to Congress that, with CAPPS II, the percentage of airplane travelers going through extra screening is expected to drop significantly from 15% that undergo it today (Carlson, 2003). Decreasing the number of false positive identifications will shorten lines at airports.

TIA, on the other hand, is a set of tools to assist agencies such as the FBI with data mining. It is designed to detect extremely rare patterns. The program will include terrorism scenarios based on previous attacks, intelligence analysis, "war games in which clever people imagine ways to attack the United States and its deployed forces," testified Anthony Tether, director of DARPA, to Congress (Carlson, 2003, p. 22). When asked how DARPA will ensure that personal information caught in TIA's net is correct, Tether stated that "we're not the people who collect the data. We're the people who supply the analytical tools to the people who collect the data" (Gross, 2003, p. 18). "Critics of data mining say that while the technology is guaranteed to invade personal privacy, it is not certain to enhance national security. Terrorists do not operate under discernable patterns, critics say, and therefore the technology will likely be targeted primarily at innocent people" (Carlson, 2003, p. 22). Congress voted to block funding of TIA. But privacy advocates are concerned that the TIA architecture, dubbed "mass dataveillance," may be used as a model for other programs (Carlson, 2003).

Systems such as TIA and CAPPS II raise a number of ethical concerns, as evidenced by the overwhelming opposition to these systems. One system, the Multistate Anti-TeRrorism Information EXchange (MATRIX), represents how data mining has a bad reputation in the public sector. MATRIX is self-defined as "a pilot effort to increase and enhance the exchange of sensitive terrorism and other criminal activity information between local, state, and federal law enforcement agencies" (matrix-at.org, accessed June 27, 2004). Interestingly, MATRIX states explicitly on its Web site that it is not a data-mining application, although the American Civil Liberties Union (ACLU) openly disagrees. At the very least, the perceived opportunity for creating ethical dilemmas and ultimately abuse is something the public is very concerned about, so much so that the project felt that the disclaimer was needed. Due to the extensive writings on data mining in the private sector, the next subsection is brief.

Ethics of Data Mining in the Private Sector

Businesses discriminate constantly. Customers are classified, receiving different services or different cost structures. As long as discrimination is not based on protected characteristics such as age, race, or gender, discriminating is legal. Technological advances make it possible to track in great detail what a person does. Michael Turner, executive director of the Information Services Executive Council, states, "For instance, detailed consumer information lets apparel retailers market their products to consumers with more precision. But if privacy rules impose restrictions and barriers to data collection, those limitations could increase the prices consumers pay when they buy from catalog or online apparel retailers by 3.5% to 11%" (Thibodeau, 2001, p. 36). Obviously, if retailers cannot target their advertising, then their only option is to mass advertise, which drives up costs.

With this profile of personal details comes a substantial ethical obligation to safeguard this data. Ignoring any legal ramifications, the ethical responsibility is placed firmly on IS professionals and businesses, whether they like it or not; otherwise, they risk lawsuits and harming individuals. "The data industry has come under harsh review. There is a raft of federal and local laws under consideration to control the collection, sale, and use of data. American companies have yet to match the tougher privacy regulations already in place in Europe, while personal and class-action litigation against businesses over data privacy issues is increasing" (Wilder & Soat, 2001, p. 38).

FUTURE TRENDS

Data mining traditionally was performed by a trained specialist, using a stand-alone package. This once nascent technique is now being integrated into an increasing number of broader business applications and legacy systems used by those with little formal training, if any, in statistics and other related disciplines. Only recently has privacy and data mining been addressed together, as evidenced by the fact that the first workshop on the subject was held in 2002 (Clifton & Estivill-Castro, 2002). The challenge of ensuring that data mining is used in an ethically and socially responsible manner will increase dramatically.

CONCLUSION

Several lessons should be learned. First, decision makers must understand key strategic issues. The data miner must have an honest and frank dialog with the sponsor concerning objectives. Second, decision makers must not come to rely on data mining to make decisions for them. The best data mining is susceptible to human interpretation. Third, decision makers must be careful not to explain away with intuition data mining results that are counterintuitive. Decision making inherently creates ethical dilemmas, and data mining is but a tool to assist management in key decisions.

REFERENCES

Adomavicius, G. & Tuzhilin, A. (2001). Using data mining methods to build customer profiles. *Computer*, *34*(2), 74-82.

Brey, P. (2000). Disclosive computer ethics. *Computers and Society, 30*(4), 10-16.

Carlson, C. (2003a). Feds look at data mining. *eWeek, 20*(19), 22.

Carlson, C. (2003b). Lawmakers will drill down into data mining. *eWeek, 20*(13), 28.

Clifton, C. & Estivill-Castro, V. (Eds.). (2002). Privacy, security and data mining. *Proceedings of the IEEE International Conference on Data Mining Workshop on Privacy, Security, and Data Mining*, Maebashi City, Japan.

Edelstein, H. (2003). Description is not prediction. *DM Review, 13*(3), 10.

Feenberg, A. (1999). *Questioning technology*. London: Routledge.

Fule, P., & Roddick, J.F. (2004). Detecting privacy and ethical sensitivity in data mining results. *Proceedings of the 27th Conference on Australasian Computer Science*, Dunedin, New Zealand.

Gross, G. (2003). U.S. agencies defend data mining plans. *ComputerWorld, 37*(19), 18.

Sclove, R. (1995). *Democracy and technology*. New York: Guilford Press.

Scott, M.D. (2002). Can data mining be a crime? *CIO Insight, 1*(10), 65.

Skalak, D. (2001). Data mining blunders exposed! 10 data mining mistakes to avoid making today. *DB2 Magazine, 6*(2), 10-13.

Thibodeau, P. (2001). FTC examines privacy issues raised by data collectors. *ComputerWorld, 35*(13), 36.

Verton, D. (2002a). Congressman says data mining could have prevented 9-11. *ComputerWorld, 36*(35), 5.

Verton, D. (2002b). Database woes thwart counterterrorism work. *ComputerWorld, 36*(49), 14.

Wilder, C., & Soat, J. (2001). The ethics of data. *Information Week, 1*(837), 37-48.

Winner, L. (1980). Do artifacts have politics? *Daedalus, 109*, 121-136.

KEY TERMS

Applied Ethics: The study of a morally controversial practice, whereby the practice is described and analyzed, and moral principles and judgments are applied, resulting in a set of recommendations.

Ethics: The study of the general nature of morals and values as well as specific moral choices; it also may refer to the rules or standards of conduct that are agreed upon by cultures and organizations that govern personal or professional conduct.

Factual Data: Data that include demographic information such as name, gender, and birth date. It also may contain information derived from transactional data such as someone's favorite beverage.

Factual Data Analysis: Another term for data mining, often used by government agencies. It uses both factual and transactional data.

Informational Privacy: The control over one's personal information in the form of text, pictures, recordings, and such.

Mass Dataveillance: Suspicion-less surveillance of large groups of people.

Relational Privacy: The control over one's person and one's personal environment.

Transactional Data: Data that contains records of purchases over a given period of time, including such information as date, product purchased, and any special requests.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 454-458, copyright 2005 by IGI Publishing, formerly known as Idea Group Publishing (an imprint of IGI Global).

Evaluation of Data Mining Methods

Paolo Giudici
University of Pavia, Italy

INTRODUCTION

Several classes of computational and statistical methods for data mining are available. Each class can be parameterised so that models within the class differ in terms of such parameters (See for instance Giudici, 2003, Hastie et al., 2001, Han and Kamber, 200, Hand et al, 2001 and Witten and Frank, 1999). For example the class of linear regression models, which differ in the number of explanatory variables; the class of bayesian networks, which differ in the number of conditional dependencies (links in the graph); the class of tree models, which differ in the number of leaves and the class multi-layer perceptrons which differ in terms of the number of hidden strata and nodes. Once a class of models has been established the problem is to choose the "best" model from it.

BACKGROUND

A rigorous method to compare models is statistical hypothesis testing. With this in mind one can adopt a sequential procedure that allows a model to be chosen through a sequence of pairwise test comparisons. However, we point out that these procedures are generally not applicable, in particular to computational data mining models, which do not necessarily have an underlying probabilistic model and, therefore, do not allow the application of statistical hypotheses testing theory. Furthermore, it often happens that for a data problem it is possible to use more than one type of model class, with different underlying probabilistic assumptions. For example, for a problem of predictive classification it is possible to use both logistic regression and tree models as well as neural networks.

We also point out that model specification and, therefore, model choice is determined by the type of variables used. These variables can be the result of transformations or of the elimination of observations, following an exploratory analysis. We then need to compare models based on different sets of variables present at the start. For example, how do we compare a linear model with the original explanatory variables with one with a set of transformed explanatory variables?

The previous considerations suggest the need for a systematic study of the methods for comparison and evaluation of data mining models.

MAIN THRUST OF THE CHAPTER

Comparison criteria for data mining models can be classified schematically into: criteria based on statistical tests, based on scoring functions, computational criteria, bayesian criteria and business criteria.

Criteria Based on Statistical Tests

The first are based on the theory of statistical hypothesis testing and, therefore, there is a lot of detailed literature related to this topic. See for example a text about statistical inference, such as Mood, Graybill and Boes (1991) and Bickel and Doksum (1977). A statistical model can be specified by a discrete probability function or by a probability density function, f(x). Such model is usually left unspecified, up to unknown quantities that have to be estimated on the basis of the data at hand. Typically, the observed sample it is not sufficient to reconstruct each detail of f(x), but can indeed be used to approximate f(x) with a certain accuracy. Often a density function is parametric so that it is defined by a vector of parameters $\Theta=(\theta_1,\ldots,\theta_I)$, such that each value θ of Θ corresponds to a particular density function, $p_\theta(x)$. In order to measure the accuracy of a parametric model, one can resort to the notion of distance between a model f, which underlies the data, and an approximating model g (see, for instance, Zucchini,

2000). Notable examples of distance functions are, for categorical variables: the entropic distance, which describes the proportional reduction of the heterogeneity of the dependent variable; the chi-squared distance, based on the distance from the case of independence; the 0-1 distance, which leads to misclassification rates. For quantitative variables, the typical choice is the Euclidean distance, representing the distance between two vectors in a Cartesian space. Another possible choice is the uniform distance, applied when nonparametric models are being used.

Any of the previous distances can be employed to define the notion of discrepancy of an statistical model. The discrepancy of a model, g, can be obtained comparing the unknown probabilistic model, f, and the best parametric statistical model. Since f is unknown, closeness can be measured with respect to a sample estimate of the unknown density f. A common choice of discrepancy function is the Kullback-Leibler divergence, that can be applied to any type of observations. In such context, the best model can be interpreted as that with a minimal loss of information from the true unknown distribution.

It can be shown that the statistical tests used for model comparison are generally based on estimators of the total Kullback-Leibler discrepancy; the most used is the log-likelihood score. Statistical hypothesis testing is based on subsequent pairwise comparisons of log-likelihood scores of alternative models. Hypothesis testing allows to derive a threshold below which the difference between two models is not significant and, therefore, the simpler models can be chosen.

Therefore, with statistical tests it is possible make an accurate choice among the models. The defect of this procedure is that it allows only a partial ordering of models, requiring a comparison between model pairs and, therefore, with a large number of alternatives it is necessary to make heuristic choices regarding the comparison strategy (such as choosing among the forward, backward and stepwise criteria, whose results may diverge). Furthermore, a probabilistic model must be assumed to hold, and this may not always be possible.

Criteria Based on Scoring Functions

A less structured approach has been developed in the field of information theory, giving rise to criteria based on score functions. These criteria give each model a score, which puts them into some kind of complete order. We have seen how the Kullback-Leibler discrepancy can be used to derive statistical tests to compare models. In many cases, however, a formal test cannot be derived. For this reason, it is important to develop scoring functions, that attach a score to each model. The Kullback-Leibler discrepancy estimator is an example of such a scoring function that, for complex models, can be often be approximated asymptotically. A problem with the Kullback-Leibler score is that it depends on the complexity of a model as described, for instance, by the number of parameters. It is thus necessary to employ score functions that penalise model complexity.

The most important of such functions is the AIC (Akaike Information Criterion, Akaike, 1974). From its definition notice that the AIC score essentially penalises the loglikelihood score with a term that increases linearly with model complexity. The AIC criterion is based on the implicit assumption that q remains constant when the size of the sample increases. However this assumption is not always valid and therefore the AIC criterion does not lead to a consistent estimate of the dimension of the unknown model. An alternative, and consistent, scoring function is the BIC criterion (Bayesian Information Criterion), also called SBC, formulated by Schwarz (1978). As can be seen from its definition the BIC differs from the AIC only in the second part which now also depends on the sample size n. Compared to the AIC, when n increases the BIC favours simpler models. As n gets large, the first term (linear in n) will dominate the second term (logarithmic in n). This corresponds to the fact that, for a large n, the variance term in the mean squared error expression tends to be negligible. We also point out that, despite the superficial similarity between the AIC and the BIC, the first is usually justified by resorting to classical asymptotic arguments, while the second by appealing to the Bayesian framework.

To conclude, the scoring function criteria for selecting models are easy to calculate and lead to a total ordering of the models. From most statistical packages we can get the AIC and BIC scores for all the models considered. A further advantage of these criteria is that they can be used also to compare non-nested models and, more generally, models that do not belong to the same class (for instance a probabilistic neural network and a linear regression model).

However, the limit of these criteria is the lack of a threshold, as well the difficult interpretability of their measurement scale. In other words, it is not easy to determine if the difference between two models is significant or not, and how it compares to another difference. These criteria are indeed useful in a preliminary exploratory phase. To examine this criteria and to compare it with the previous ones see, for instance, Zucchini (2000), or Hand, Mannila and Smyth (2001).

Bayesian Criteria

A possible "compromise" between the previous two criteria is the Bayesian criteria which could be developed in a rather coherent way (see e.g. Bernardo and Smith, 1994). It appears to combine the advantages of the two previous approaches: a coherent decision threshold and a complete ordering. One of the problems that may arise is connected to the absence of a general purpose software. For data mining works using Bayesian criteria the reader could see, for instance, Giudici (2001), Giudici and Castelo (2003) and Brooks et al. (2003).

Computational Criteria

The intensive wide spread use of computational methods has led to the development of computationally intensive model comparison criteria. These criteria are usually based on using dataset different than the one being analysed (external validation) and are applicable to all the models considered, even when they belong to different classes (for example in the comparison between logistic regression, decision trees and neural networks, even when the latter two are non probabilistic). A possible problem with these criteria is that they take a long time to be designed and implemented, although general purpose softwares have made this task easier.

The most common of such criterion is based on cross-validation. The idea of the cross-validation method is to divide the sample into two sub-samples, a "training" sample, with n - m observations, and a "validation" sample, with m observations. The first sample is used to fit a model and the second is used to estimate the expected discrepancy or to assess a distance. Using this criterion the choice between two or more models is made by evaluating an appropriate discrepancy function on the validation sample. Notice that the cross-validation idea can be applied to the calculation of any distance function.

One problem regarding the cross-validation criterion is in deciding how to select m, that is, the number of the observations contained in the "validation sample". For example, if we select m = n/2 then only n/2 observations would be available to fit a model. We could reduce m but this would mean having few observations for the validation sampling group and therefore reducing the accuracy with which the choice between models is made. In practice proportions of 75% and 25% are usually used, respectively for the training and the validation samples.

To summarise, these criteria have the advantage of being generally applicable but have the disadvantage of taking a long time to be calculated and of being sensitive to the characteristics of the data being examined. A way to overcome this problem is to consider model combination methods, such as bagging and boosting. For a thorough description of these recent methodologies, see Hastie, Tibshirani and Friedman (2001).

Business Criteria

One last group of criteria seem specifically tailored for the data mining field. These are criteria that compare the performance of the models in terms of their relative losses, connected to the errors of approximation made by fitting data mining models. Criteria based on loss functions have appeared recently, although related ideas are known since longtime in Bayesian decision theory (see for instance Bernardo and Smith, 1984) . They have a great application potential although at present they are mainly concerned with classification problems. For a more detailed examination of these criteria the reader can see for example Hand (1997), Hand, Mannila and Smyth (2001), or the reference manuals on data mining software, such as that of SAS Enterprise Miner (SAS Institute, 2004).

The idea behind these methods is to focus the attention, in the choice among alternative models, to the utility of the obtained results. The best model is the one that leads to the least loss.

Most of the loss function based criteria are based on the confusion matrix. The confusion matrix is used as an indication of the properties of a classification rule. On its main diagonal it contains the number of observations that have been correctly classified for each class. The off-diagonal elements indicate the number of observations that have been incorrectly classified. If it is assumed that each incorrect classification has the

same cost, the proportion of incorrect classifications over the total number of classifications is called rate of error, or misclassification error, and it is the quantity which must be minimised. The assumption of equal costs can be replaced by weighting errors with their relative costs.

The confusion matrix gives rise to a number of graphs that can be used to assess the relative utility of a model, such as the Lift Chart, and the ROC Curve (see Giudici, 2003). The lift chart puts the validation set observations, in increasing or decreasing order, on the basis of their score, which is the probability of the response event (success), as estimated on the basis of the training set. Subsequently, it subdivides such scores in deciles. It then calculates and graphs the observed probability of success for each of the decile classes in the validation set. A model is valid if the observed success probabilities follow the same order (increasing or decreasing) as the estimated ones. Notice that, in order to be better interpreted, the lift chart of a model is usually compared with a baseline curve, for which the probability estimates are drawn in the absence of a model, that is, taking the mean of the observed success probabilities.

The ROC (Receiver Operating Characteristic) curve is a graph that also measures predictive accuracy of a model. It is based on four conditional frequencies that can be derived from a model, and the choice of a cut-off points for its scores: a) the observations predicted as events and effectively such (sensitivity); b) the observations predicted as events and effectively non events; c) the observations predicted as non events and effectively events; d) the observations predicted as non events and effectively such (specificity). The ROC curve is obtained representing, for any fixed cut-off value, a point in the plane having as x-value the false positive value (1-specificity) and as y-value the sensitivity value. Each point in the curve corresponds therefore to a particular cut-off . In terms of model comparison, the best curve is the one that is leftmost, the ideal one coinciding with the y-axis.

To summarise, criteria based on loss functions have the advantage of being easy to understand but, on the other hand, they still need formal improvements and mathematical refinements.

FUTURE TRENDS

It is well known that data mining methods can be classified into exploratory, descriptive (or unsupervised), predictive (or supervised) and local (see e.g. Hand et al, 2001). Exploratory methods are preliminary to others and, therefore, do not need a performance measure. Predictive problems, on the other hand, are the setting where model comparison methods are most needed, mainly because of the abundance of the models available. All presented criteria can be applied to predictive models: this is a rather important aid for model choice.

For descriptive models aimed at summarising variables, such as clustering methods, the evaluation of the results typically proceeds on the basis of the euclidean distance, leading at the R^2 index. We remark that it is important to examine the ratio between the "between" and "total" sums of squares, that leads to R^2 , for each variable in the dataset. This can give a variable-specific measure of the goodness of the cluster representation.

Finally, it is difficult to assess local models, such as association rules, for the bare fact that a global measure of evaluation of such model contradicts with the very notion of a local model. The idea that prevails in the literature is to measure the utility of patterns in terms of how interesting or unexpected they are to the analyst. As measures of interest one can consider, for instance, the support, the confidence and the lift. The former can be used to assess the importance of a rule, in terms of its frequency in the database; the second can be used to investigate possible dependences between variables; finally the lift can be employed to measure the distance from the situation of independence.

CONCLUSION

The evaluation of data mining methods requires a great deal of attention. A valid model evaluation and comparison can improve considerably the efficiency of a data mining process. We have presented several ways to perform model comparison, each has its advantages and disadvantages. Choosing which of them is most suited for a particular application depends on the specific problem and on the resources (e.g. comput-

ing tools and time) available. Furthermore, the choice must also be based upon the kind of usage of the final results. This implies that, for example, in a business setting, comparison criteria based on business quantities are extremely useful.

REFERENCES

Akaike, H. (1974). A new look at statistical model identification. *IEEE Transactions on Automatic Control*, 19, 716—723.

Bernardo, J.M. and Smith, A.F.M. (1994). *Bayesian Theory*, New York: Wiley.

Bickel, P.J. and Doksum, K. A. (1977). *Mathematical Statistics*, New York: Prentice Hall.

Brooks, S.P., Giudici, P. and Roberts, G.O. (2003). Efficient construction of reversible jump MCMC proposal distributions. *Journal of The Royal Statistical Society series B*, 1, 1-37..

Castelo, R. and Giudici, P. (2003). Improving Markov Chain model search for data mining. *Machine learning*, 50, 127-158.

Giudici, P. (2001). Bayesian data mining, with application to credit scoring and benchmarking. *Applied Stochastic Models in Business and Industry*, 17, 69-81.

Giudici, P. (2003). *Applied data mining*, London, Wiley.

Hand, D.J., Mannila, H. and Smyth, P (2001). *Principles of Data Mining*, New York: MIT Press.

Hand, D. (1997). *Construction and assessment of classification rules.* London: Wiley.

Han, J. and Kamber, M (2001). *Data mining: concepts and techniques*. New York: Morgan and Kaufmann.

Hastie, T., Tibshirani, R., Friedman, J. (2001). *The elements of statistical learning: data mining, inference and prediction*, New York: Springer-Verlag.

Mood, A.M., Graybill, F.A. and Boes, D.C. (1991). *Introduction to the theory of Statistics*, Tokyo: Mc Graw Hill.

SAS Institute Inc. (2004). *SAS enterprise miner reference manual*. Cary: SAS Institute.

Schwarz, G. (1978). *Estimating the dimension of a model*. *Annals of Statistics* 62, 461-464.

Witten, I. and Frank, E. (1999). *Data Mining: practical machine learning tools and techniques with Java implementation.* New York: Morgan and Kaufmann.

Zucchini, Walter (2000). An Introduction to Model Selection. *Journal of Mathematical Psychology*, 44, 41-61.

KEY TERMS

Entropic Distance: The entropic distance of a distribution g from a target distribution f, is:

$$_E d = \sum_i f_i \log \frac{f_i}{g_i}$$

Chi-Squared Distance: The chi-squared distance of a distribution g from a target distribution f is:

$$_{\chi^2} d = \sum_i \frac{(f_i - g_i)^2}{g_i}$$

0-1 Distance: The 0-1 distance between a vector of predicted values, X_g, and a vector of observed values, X_f, is:

$$_{0-1} d = \sum_{r=1}^{n} 1(X_{fr} - X_{gr})$$

where 1(w,z) =1 if w=z and 0 otherwise.

Euclidean Distance: the distance between a a vector of predicted values, X_g, and a vector of observed values, X_f, is expressed by the equation:

$$_2 d(X_f, X_g) = \sqrt{\sum_{r=1}^{n} (X_{fr} - X_{gr})^2}$$

Uniform Distance: The uniform distance betweeen two distribution functions, F and G, with values in [0, 1] is defined by:

$$\sup_{0 \le t \le 1} |F(t) - G(t)|$$

Discrepancy of a Model: Assume that f represents the unknown density of the population, and let g= p_θ be a family of density functions (indexed by a vector of I parameters, θ) that approximates it. Using, to exemplify, the Euclidean distance, the discrepancy of a model g, with respect to a target model f is:

$$\Delta(f, p_\vartheta) = \sum_{i=1}^{n} \left(f(x_i) - p_\vartheta(x_i)\right)^2$$

Kullback-Leibler Divergence: The Kullback-Leibler divergence of a parametric model p_θ with respect to an unknown density f is defined by:

$$\Delta_{K-L}(f, p_\vartheta) = \sum_i f(x_i) \log\left(\frac{f(x_i)}{p_{\hat{\theta}}(x_i)}\right)$$

where the suffix $\hat{\theta}$ indicates the values of the parameters which minimizes the distance with respect to f.

Log-Likelihood Score: The log-likelihood score is defined by

$$-2\sum_{i=1}^{n} \log\left[p_{\hat{\theta}}(x_i)\right]$$

AIC Criterion: The AIC criterion is defined by the following equation:

$$AIC = -2\log L(\hat{\vartheta}; x_1, ..., x_n) + 2q$$

where $\log L(\hat{\vartheta}; x_1, ..., x_n)$ is the logarithm of the likelihood function calculated in the maximum likelihood parameter estimate and q is the number of parameters of the model.

BIC Criterion: The BIC criterion is defined by the following expression:

$$BIC = -2\log L\left(\hat{\vartheta}; x_1, ..., x_n\right) + q\log(n)$$

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 464-468, copyright 2005 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

Evaluation of Decision Rules by Qualities for Decision-Making Systems

Ivan Bruha
McMaster University, Canada

INTRODUCTION

A 'traditional' learning algorithm that can induce a set of decision rules usually represents a robust and comprehensive system that discovers a knowledge from usually large datasets. We call this discipline Data Mining (DM). Any classifier, expert system, or generally a decision-supporting system can then utilize this decision set to derive a decision (prediction) about given problems, observations, diagnostics. DM can be defined as a nontrivial process of identifying valid, novel, and ultimately understandable knowledge in data. It is understood that DM as a multidisciplinary activity points to the overall process of determining a useful knowledge from databases, i.e. extracting high-level knowledge from low-level data in the context of large databases.

A rule-inducing learning algorithm may yield either an ordered or unordered set of *decision rules*. The latter seems to be more understandable by humans and directly applicable in most expert systems or decision-supporting ones. However, classification utilizing the unordered-mode decision rules may be accompanied by some conflict situations, particularly when several rules belonging to different classes match ('fire' for) an input to-be-classified (unseen) object. One of the possible solutions to this conflict is to associate each decision rule induced by a learning algorithm with a numerical factor which is commonly called the *rule quality*.

The chapter first surveys empirical and statistical formulas of the rule quality and compares their characteristics. Statistical tools such as contingency tables, rule consistency, completeness, quality, measures of association, measures of agreement are introduced as suitable vehicles for depicting a behaviour of a decision rule.

After that, a very brief theoretical methodology for defining rule qualities is acquainted. The chapter then concludes by analysis of the formulas for rule qualities, and exhibits a list of future trends in this discipline.

BACKGROUND

Machine Learning (ML) or Data Mining (DM) utilize several paradigms for extracting a knowledge that can be then exploited as a decision scenario (architecture) within an expert system, classification (prediction) one, or any decision-supporting one. One commonly used paradigm in Machine Learning is *divide-and-conquer* that induces decision trees (Quinlan, 1994). Another widely used *covering* paradigm generates sets of decision rules, e.g., the CNx family (Clark & Boswell, 1991; Bruha, 1997), C4.5Rules and Ripper. However, the rule-based classification systems are faced by an important deficiency that is to be solved in order to improve the predictive power of such systems; this issue is discussed in the next section.

Also, it should be mentioned that they are two types of agents in the multistrategy decision-supporting architecture. The simpler one yields a single decision; the more sophisticated one induces a list of several decisions. In both types, each decision should be accompanied by the agent's confidence (belief) into it. These functional measurements are mostly supported by statistical analysis that is based on both the certainty (accuracy, predictability) of the agent itself as well as consistency of its decision. There have been quite a few research enquiries to define formally such statistics; some, however, have yielded in quite complex and hardly enumerable formulas so that they have never been used.

One of the possible solutions to solve the above problems is to associate each decision rule induced by a learning algorithm with a numerical factor: a *rule quality*. The issue fo the rule quality was discussed in many papers; here we introduce just the most essential ones: (Bergadano et al., 1988; Mingers, 1989) were evidently one of the first papers introducing this problematic. (Kononenko, 1992; Bruha, 1997) were the followers; particularly the latter paper presented a methodological insight to this discipline. (An & Cercone,

2001) just extended some of the techniques introduces by (Bruha,1997). (Tkadlec & Bruha, 2003) presents a theoretical methodology and general definitions of the notions of a Designer, Learner, and Classifier in a formal matter, including parameters that are usually attached to these concepts such as rule consistency, completeness, quality, matching rate, etc. That paper also provides the minimum-requirement definitions as necessary conditions for the above concepts. Any designer (decision-system builder) of a new multiple-rule system may start with these minimum requirements.

RULE QUALITY

A rule-inducing algorithm may yield either an ordered or unordered set of decision rules. The latter seems to be more understandable by humans and directly applicable in most decision-supporting systems. However, the classification utilizing an unordered set of decision rules exhibits a significant deficiency, not immediately apparent. Three cases are possible:

1. If an input unseen (to-be-classified) object satisfies (matches, 'fires' for) one or more rules of the same class, then the object is categorized to the class assigned to the rule(s).
2. If the unseen object is not covered by any rule, then either the classifier informs the user about its inability to decide ('I do not know'), or the object is assigned by default to the majority class in the training set, or some similar techniques are invoked.
3. Difficulty arises if the input object satisfies more rules assigned to different classes. Then some schemes have to be applied to assign the unseen input object to the most appropriate class.

One possibility to clarify the conflict situation (case 3) of multiple-rule systems is to associate each rule in the decision-supporting scheme with a numerical factor that can express its properties and characterize a measure of belief in the rule, its power, predictability, reliability, likelihood, and so forth. A collection of these properties is symbolized by a function commonly called the *rule quality*. After choosing a formula for the rule quality, we also have to select a scheme for combining these qualities (*quality combination*).

Quality of rules, its methodology as well as appropriate formulas have been discussed for many years. (Bergadano et al., 1992) is one the first papers that introduces various definitions and formulas for the rule quality; besides rule's power and predictability it measures its size, understandability, and other factors. A survey of the rule combinations can be found, e.g. in (Kohavi & Kunz, 1997). Comprehensive analysis and empirical expertise of formulas of rule qualities and their combining schemes has been published in (Bruha & Tkadlec, 2003), its theoretical methodology in (Tkadlec & Bruha, 2003).

We now discuss the general characteristics of a formula of the rule quality. The first feature required for the rule quality is its *monotony* (or, more precisely, *nondecreasibility*) towards its arguments. Its common arguments are the *consistency* and *completeness* factors of decision rules. Consistency of a decision rule exhibits its 'purity' or reliability, i.e., a rule with high consistency should cover the minimum of the objects that do not belong to the class of the given rule. A rule with high completeness factor, on the other hand, should cover the maximum of objects belonging to the rule's class.

The reason for exploiting the above characteristics is obvious. Any DM algorithm dealing with real-world noisy data is to induce decision rules that cover larger numbers of training examples (objects) even with a few negative ones (not belonging to the class of the rule). In other words, the decision set induced must be not only reliable but also powerful. Its reliability is characterized by a consistency factor and its power by a completeness factor.

Besides the rule quality discussed above there exist other rule measures such as its size (e.g., the size of its condition, usually the number of attribute pairs forming the condition), computational complexity, comprehensibility ('Is the rule telling humans something interesting about the application domain?'), understandability, redundancy (measured within the entire decision set of rules), and similar characteristics (Tan et al., 2002; Srivastava, 2005). However, some of these characteristics are subjective; on contrary, formulas of rule quality are supported by theoretical sources or profound empirical expertise.

Here we just briefly survey the most important characteristics and definitions used by the formulas of rule qualities follow. Let a given task to be classified be characterized by a set of training examples that belong

to two classes, named C and $\bar{C}$ (or, not C). Let R be a decision rule of the class C, i.e.

`R: if Cond then class is C`

Here `R` is the name of the rule, `Cond` represents the condition under which the rule is satisfied (fires), and `C` is the class of the rule, i.e., an unseen object satisfying this condition is classified to the class `C`.

Behaviour of the above decision rule can be formally depicted by the 2×2 *contingency table* (Bishop et al., 1991), which is commonly used in machine learning (Mingers, 1989):

	class C	not class C	
rule R covers	a_{11}	a_{12}	a_{1+}
R does not cover	a_{21}	a_{22}	a_{2+}
	a_{+1}	a_{+2}	a_{++}

where

a_{11} is the number of training examples that are covered by (satisfied by) the rule R and belong to the class C,

a_{12} is the number of examples covered by (satisfied by) the rule R but not belonging to the class C, etc.,

$a_{1+} = a_{11} + a_{12}$ is the number of examples covered by R,

$a_{+1} = a_{11} + a_{21}$ is the number of the training examples of the class C,

$a_{++} = a_{+1} + a_{+2} = a_{1+} + a_{2+}$ is the number of all training examples in the given task.

Using the elements of the contingency table, we may define the *consistency* (*sensitivity*) of a rule R

$$\text{cons}(R) = \frac{a_{11}}{a_{1+}}$$

and its *completeness* (*coverage*) by

$$\text{compl}(R) = \frac{a_{11}}{a_{+1}}$$

Note that other statistics can be easily defined by means of the elements of the above contingency table. The conditional probability $P(C|R)$ of the class C under the condition that the rule R matches an input object is, in fact, equal to cons(R) ; similarly, the probability $P(R|C)$ that the rule R fires under the condition the input object is from the class C is identical to the rule's completeness. The prior probability of the class C is

$$P(C) = \frac{a_{+1}}{a_{++}}$$

Generally we may state that these formulas are functions of any above characteristics, i.e. all nine elements a_{11} to a_{++} of the contingency table, consistency, completeness, and various probabilities above classes and rules. There exist two groups of these formulas: empirical and statistical ones.

1. Empirical Formulas

The way of defining rule quality is an interesting issue in machine learning (Bergadano et al., 1988; Brazdil & Torgo, 1990; Torgo, 1993). However, the majority of these formulas represent an empirical, ad-hoc approach to the definition of rule quality, because they are based on intuitive logic and not necessarily backed by statistical or other theories. The common strategy is to use a weighted sum or multiplication of the consistency and completeness, which is more understandable to humans.

Rule quality as a *weighted sum* of the consistency and completeness has the form:

$$q^{\text{weight}}(R) = w_1 \ \text{cons}(R) + w_2 \ \text{compl}(R)$$

where $w_1, w_2 \in (0,1)$ are user-defined weights, usually $w_1 + w_2 = 1$. This apprehensible formula is nondecreasing towards its arguments. (Torgo, 1993) specifies the above weights by a heuristic formula that emphasizes consistency:

$$w_1 = 0.5 + \tfrac{1}{4}\,\text{cons}(R)$$
$$w_2 = 0.5 \,!\, \tfrac{1}{4}\,\text{cons}(R)$$

Rule quality as a *product* of consistency and completeness has the form:

$$q^{\text{product}}(R) = \text{cons}(R) \,.\, f\,(\text{compl}(R))$$

where f is an increasing function. The completeness works here as a factor that reflects a *confidence* to the consistency of the given rule. After a large number of experiments, (Brazdil & Torgo, 1990) selected an

empiric form of the function f: $f(x) = \exp(x - 1)$

2. Statistical Formulas

The statistical formulas for rule quality are supported by a few theoretical sources. Here we survey two of them.

2A. The theory on ***contingency tables*** seems to be one of these sources, since the performance of any rule can be characterized by them. Generally, there are two groups of measurements (see e.g. (Bishop et al.,1991)): association and agreement.

A measure of *association* (Bishop et al., 1991) indicates a relationship between rows and columns in a 2×2 contingency table so that the *i*-th row may be 'associated' with the *j*-th column. In other words, this measure reflects an 'association' on both diagonals of the contingency table. Two following measures may serve as reasonable rule qualities: the Pearson χ^2 statistic applied to the 2×2 contingency table

$$q^{X^2}(R) = \frac{(a_{11}a_{22} - a_{12}a_{21})^2}{a_{+1}\,a_{+2}\,a_{1+}\,a_{2+}}$$

and the *G2*-likelihood statistic

$$q^{G2}(R) = 2\left(a_{11}\ln\frac{a_{11}a_{++}}{a_{1+}a_{+1}} + a_{12}\ln\frac{a_{12}a_{++}}{a_{1+}a_{+2}}\right).$$

It should be noted that the *J*-measure introduced in (Smyth & Goodman, 1990) and often quoted as one of the most promising measures (Kononenko, 1992), in fact, is equal to the *G2*-likelihood statistic divided by $2a_{++}$.

From the viewpoint of machine learning, the above formulas work properly only if the class *C* is the majority class of examples covered by the rule.

A measure of *agreement* is a special case of association that indicates an 'association' of the elements of a contingency table on its main diagonal only. The simplest measure proposed in (Bishop et al., 1991) is the sum of the main diagonal $a_{11} + a_{22}$.

(Cohen, 1960) suggests comparison of the *actual* agreement

$$Aagree = \frac{1}{a_{++}}\sum_i a_{ii}$$

with the *chance* agreement

$$Cagree = \frac{1}{a_{++}^2}\sum_i a_{i+}a_{+i}$$

which occurs if the row variable is independent of the column variable, i.e., if the rule's coverage does not yield any information about the class *C*. The difference *Aagree* - *Cagree* is then normalized by its maximum possible value. This leads to the measure of agreement that we interpret as a rule quality (named after its author):

$$q^{\text{Cohen}}(R) = \frac{\frac{1}{a_{++}}\sum_i a_{ii} - \frac{1}{a_{++}^2}\sum_i a_{i+}a_{+i}}{1 - \frac{1}{a_{++}^2}\sum_i a_{i+}a_{+i}}$$

Note that here the coefficients $1/a_{++}$ and $1/a_{++}^2$ are used because the a_{ij}'s are *absolute*, not relative frequencies.

Cohen's statistic provides a formula which responds to the agreement on the main diagonal of the contingency table, i.e. when both components a_{11} and a_{22} are reasonably large. On the other hand, Coleman (Bishop, 1991) defines a measure of agreement that indicates an 'association' between the first column and any particular row in the contingency table. For the purpose of the rule quality definition, the agreement between the first column ('example from the class *C* ') and the first row ('rule *R* covers an example') is the proper one. The formula follows Cohen's statistic in principle by normalizing the difference between the 'actual' and 'chance' agreement. Hence, Coleman's formula for the rule quality is

$$q^{\text{Coleman}}(R) = \frac{a_{11} - \frac{1}{a_{++}}a_{1+}a_{+1}}{a_{1+} - \frac{1}{a_{++}}a_{1+}a_{+1}}$$

Coleman's statistic intuitively exhibits necessary properties of a rule quality: agreement between 'rule covers' and 'example from class *C* '. Nevertheless, one can find that Coleman's formula does not properly comprise the completeness of a rule. Therefore, (Bruha & Tkadlec, 2003) have changed slightly the above formula and got two modified ones that exhibit the proper characteristics.

2B. **Information theory** is another source of statistical measurements suitable for defining rule quality.

Following this theory, (Kononenko & Bratko, 1991) define the *information score* for a rule R when classifying an unseen object as $-\log_2 P(C) + \log_2 P(C|R)$. This formula can be also applied to the rule quality:

$$q^{\mathrm{IS}}(R) = -\log_2 \frac{a_{+1}}{a_{++}} + \log_2 \mathrm{cons}(R)$$

More precisely, this formula is valid only if $\mathrm{cons}(R) \geq \frac{a_{+1}}{a_{++}}$ which could be interpreted as a necessary condition for the reliability of a rule. This quality is evidently an increasing function towards its arguments.

The *logical sufficiency* can be also applied for defining a rule quality. Given a rule R and a class C, the logical sufficiency of the rule R with respect to C is defined as

$$q^{\mathrm{LS}}(R) = \frac{P(R \mid C)}{P(R \mid \overline{C})} = \frac{a_{11}a_{+2}}{a_{12}a_{+1}}$$

where $\overline{C}$ represents 'not class C '. A large value of the rule quality $q^{\mathrm{LS}}(R)$ means that the fired rule R supports the class C. Slightly modified formula called *discrimination* can be used for the rule quality, as well:

$$q^{\mathrm{D}}(R) = \frac{P(R \mid C)(1 - P(R \mid \overline{C}))}{P(R \mid \overline{C})(1 - P(R \mid C))} = \frac{a_{11}a_{22}}{a_{12}a_{21}}$$

Its purpose is to find a rule which can discriminate between positive and negative examples of the class C.

This section just discusses, introduces, and analyzes a few empirical and statistical formulas for the rule quality. This direction in DM and ML, however, would need a theoretical support and methodological background. (Tkadlec & Bruha, 2003) expresses just the first step in this direction. Their paper presents a four-tuple of Designer, Learner, Classifier, and Predictor; besides the rule quality, it also introduces the concept of importances (matching ratios) both cumulated (over-all) and class-sensitive. Last but not least, it provides a theoretical methodology for the designer (domain expert, decision-system builder) how to build up a multiple-rule system.

FUTURE TRENDS

Here are some directions the research of the evaluation of decision-supporting systems can continue.

- When we choose a formula for the rule quality, we also have to select a *scheme for combining* these qualities. It may happen that an unseen (to-be-classified) object matches (satisfies) more rules that belong to different classes; to solve this conflict case, the qualities of fired rules of the same class have to be combined using a certain scheme (formula). Consequently, the rule-quality combination with the maximum value will determine the class of the unseen object. There exist just preliminary studies of the combination schemes, e.g., (Kononenko, 1992). One direction in the future research should thus invent a few powerful and robust combination schemes, incl. the corresponding methodology and theory.
- Also, the research is to continue in formulating more sophisticated methodology and theory above the problematic of designing new formulas for rule qualities. (Tkadlec & Bruha, 2003) is just the first step; the paper, in fact, does not exhibit any profound theorem, just methodology, definitions, and a couple of lemmas.
- The methodology and theoretical formalism for single decision rules can be extended to the entire decision set of rules (knowledge base, model). A *model quality* thus represents the predictive power of the entire model (decision set of rules). It can be consequently utilized in a multi-level decision-supporting system which consists of two levels: the base level (consisting of 'base' models) and the second one (a 'super-model', meta-model). This second level then combines the decisions of the base models (utilizing their model qualities) in order to make up the final decision. (We can view a base model as a 'physician' who has to find a diagnosis of a patient, and the second level as a 'council' of physicians that combines the decisions of all the members according to physicians' qualities, and makes up the final verdict of the patient's diagnosis.)
- In most decision-supporting systems, the rule qualities are static, constant, calculated a priori, before the actual classification or prediction. Their

predictability can be improved by a dynamic change of their values during the classification process. One possible scheme implants a feedback loop from the classifier to the learner (Bruha, 2000); it refines (modifies) the rule qualities according to the correct/false predictions made by the classifier by changing the qualities of the rules that were involved in the current classification.

CONCLUSION

This chapter deals with the conceptuality of rule quality, and solves some conflicts in multiple-rule systems. Many experimental comparisons of various formulas of rule quality has been carried out in several papers, e.g., (Bruha & Tkadlec, 2003; An & Cercone, 2001).

By analyzing those experiments, one can find that the weighted sum - particularly with the weights proposed by (Torgo, 1993) - and the two modifications of Coleman's quality exhibit the best performance. The second successful group (as for the classification accuracy) are q^{product}, q^{LS}, and the *J*-measure. The worst rule qualities are the formulas q^{IS} and q^{χ^2}.

The weighted-sum quality exhibits quite interesting characteristic. By selecting the weights w_1 and w_2 a user may choose between inducing only reliable rules (when supporting consistency of rules) and only robust rules (when supporting completeness). However, our experiments undoubtedly revealed that a larger support of completeness deteriorates the classification accuracy.

Generally speaking, the statistical formulas introduced above exhibit the *independence* property: they yield the value zero if and only if the rule does not have any expressive capability, i.e., if its distribution is equal to the distribution of the entire training set. The name is derived from the fact that the columns and rows in the contingency table are independent in this case. The independence property can be also expressed by the consistency or completeness; it arises if

$$\text{cons}(R) = \text{compl}(R) = \frac{a_{+1}}{a_{++}}$$

One may observe that majority of the quality formulas introduced here are applicable. Although the empirical formulas are not backed by any statistical theory, they work quite well. Also, the classification accuracy of the statistical formulas has negligible difference in comparison to the empirical ones. On the other hand, statistical formulas have been derived from statistical theories and exhibit the independence property which has the advantage that the threshold of meaningful rules is clearly expressed. To be more precise, even the statistical formulas are applied mostly in an ad-hoc, empirical fashion, since most algorithms do not check the conditions or assumptions of their applicability. Unlike the empirical formulas, however, we are aware of the error we have made when the conditions of applicability are not checked. If needed, a checking condition could in principle be embedded into the algorithm.

REFERENCES

An, A. & Cercone, N. (2001). Rule Quality Measures for Rule Induction Systems: Description and Evaluation. *Computational Intelligence J.*

Bergadano, F. et al. (1988). Measuring quality of concept descriptions. *EWSL-88*, Glasgow, 112-128.

Bergadano, F. et al. (1992). Learning Two-Tiered Descriptions of Flexible Concepts: The Poseidon system. *Machine Learning*, 8, 5-43.

Bishop, Y.M.M., Fienberg, S.E., & Holland, P.W. (1991). *Discrete multivariate analysis: Theory and practice*, The MIT Press.

Brazdil, P. & Torgo, L. (1990). Knowledge Acquisition via Knowledge Integration. In: *Current Trends in Knowledge Acquisition*, IOS Press.

Bruha, I. (1997). Quality of decision rules: Definitions and classification schemes for multiple rules. In: Nakhaeizadeh, G. & Taylor, C.C. (Eds.), *Machine Learning and Statistics: The Interface*, John Wiley, New York, 107-131.

Bruha, I. (2000). A Feedback Loop for Refining Rule Qualities in a Classifier: A Reward-Penalty Strategy. *European Conference on Machine Learning (ECML-2000), Workshop Meta Learning*, 15-27.

Bruha, I. & Tkadlec, J. (2003). Rule Quality for Multiple-rule Classifier: Empirical Expertise and Theoretical Methodology. *Intelligent Data Analysis J.*, **7**, 99-124.

Clark, P. & Boswell, R. (1991). Rule Induction with CN2: Some Recent Improvements. *EWSL-91*, Porto.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psych. Meas.*, 20, 37-46.

Kohavi, R. & Kunz, C. (1997). Optional Decision Trees with Majority Votes. In: Fisher, D. (ed.): *Machine Learning: Proc. 14th International Conference*, Morgan Kaufmann, 161-169.

Kononenko, I. & Bratko, I. (1991). Information-based evaluation criterion for classifier's performance. *Machine Learning*, 6, 67-80.

Kononenko, I. (1992). Combining decisions of multiple rules. In: du Boulay, B. & Sgurev, V. (Eds.), *Artificial Intelligence V: Methodology, Systems, Applications*, Elsevier Science Publ., 87-96.

Mingers, J. (1989). An empirical comparison of selection measures for decision-tree induction. *Machine Learning*, 3, 319-342.

Quinlan, J.R. (1994). *C4.5: Programs for machine learning*. Morgan Kaufmann.

Smyth, P. & Goodman, R.M. (1990). Rule induction using information theory. In: Piarersky, P. & Frawley, W. (Eds.), *Knowledge Discovery in Databases*, MIT Press.

Srivastava,M.S. (2005). Some Tests Concerning The Covariance Matrix In High Dimensional Data. *Journal of Japan Statistical Soc.*, 35, 251-272.

Tkadlec, J. & Bruha, I. (2003). Formal Aspects of a Multiple-Rule Classifier. *International J. Pattern Recognition and Artificial Intelligence*, 17, 4, 581-600.

Torgo, L. (1993). Controlled redundancy in incremental rule learning. *ECML-93*, Vienna, 185-195.

Tan, P.N., Kumar, V., & Srivastava, J. (2002). Selecting the Right Interestingness Measure for Association Patterns. *Proc. of the Eighth ACM SIGKDD International Conf. on Knowledge Discovery and Data Mining (SIGKDD-2002)*, 157-166.

KEY TERMS

Classifier: A decision-supporting system that given an unseen input object yields a prediction, for instance, it classifies the given object to a certain class.

Completeness of a Rule: It exhibits the rule's power, i.e. a rule with high completeness factor covers the maximum of objects belonging to the rule's class.

Consistency of a Rule: It exhibits the rule's 'purity' or reliability, i.e., a rule with high consistency should cover the minimum of the objects that do not belong to the class of the given rule.

Contingency Table: A matrix of size $M*N$ whose elements represent the relation between two variables, say X and Y, the first having M discrete values X_1,..., X_M, the latter N values Y_1,..., Y_N; the (m,n)-th element of the table thus exhibits the relation between the values X_m and Y_n.

Decision Rule: An element (piece) of knowledge, usually in the form of 'if-then statement':

```
if Condition then Action/Class
```

If its `Condition` is satisfied (i.e., matches a fact in the corresponding database of a given problem, or an input object), then its `Action` (e.g., decision making) is performed, usually an input unseen (to-be-classified) object is assigned to the `Class`.

Decision Set: Ordered or unordered set of decision rules; a common knowledge representation tool (utilized e.g. in most expert systems).

Model (Knowledge Base): Formally described concept of a certain problem; usually represented by a set of production rules, decision rules, semantic nets, frames.

Model Quality: Similar to rule quality, but it characterizes the decision power, predictability, and reliability of the entire model (decision set of rules, knowledge base) as a unit.

Rule Quality: A numerical factor that characterizes a measure of belief in the given decision rule, its power, predictability, reliability, likelihood.

The Evolution of SDI Geospatial Data Clearinghouses

Maurie Caitlin Kelly
The Pennsylvania State University, USA

Bernd J. Haupt
The Pennsylvania State University, USA

Ryan E. Baxter
The Pennsylvania State University, USA

INTRODUCTION

Geospatial data and the technologies that drive them have altered the landscape of our understanding of the world around us. The data, software and services related to geospatial information have given us the opportunity to visualize existing phenomena, to understand connections, and to address problems from environmental management to emergency response. From the ever-present Google Earth images we are shown in our televised weather reports to the 3D flyovers of war zones on the news, geospatial information is everywhere. In the decade or so since U.S. President William Clinton set the stage by announcing the establishment of the National Spatial Data Infrastructure (NSDI), the concept of the geospatial data clearinghouse has shifted dramatically to fulfill the increasing need to streamline government processes, increase collaboration, and to meet the demands of data users and data developers (Clinton, 1994). The announcement of the NSDI gave birth to a Global Spatial Data Infrastructure (GSDI) movement that would be supported by a network of SDIs or geospatial data clearinghouses from local, state, and national levels.

From this point on, the evolution of the geospatial data clearinghouse has been rapid and punctuated with challenges to both the developer and the user. From the earliest incarnations of these now pervasive resources as simple FTP data transfer sites to the latest developments in Internet Map Services and real time data services, geospatial data clearinghouses have provided the backbone for the exponential growth of Geographic Information Systems (GIS). In this section, the authors will examine the background of the geospatial data clearinghouse movement, address the basic phases of clearinghouse development, and review the trends that have taken the world's clearinghouses from FTP to Internet Map Services and beyond.

THE SPATIAL DATA INFRASTRUCTURE MOVEMENT

No discussion of SDIs and geospatial data clearinghouses would be complete without a brief introduction to the history of the movement.

The growth of geospatial data clearinghouse movement can trace its origins to the spatial data infrastructure initiatives of the 1990s when spatial data sharing began in earnest. In the United States an effort to organize spatial data and develop standards for sharing data began as the NSDI. First envisioned in 1993, the concept of the coordinated data model set forth the ideas and goals of widespread sharing of data and resources (National Research Council, 1993). By 1995, the United States had developed a plan for data sharing and established a gateway by which participants could register their metadata holdings through a centralized source (FGDC95). Sharing data through this gateway required developing metadata to an accepted standard and utilized the Z39.50 protocol—both of which will be described in the next section.

The spatial data infrastructure concept as it has evolved has, at its core, the premise that sharing data eliminates redundancy, enhances opportunities for cooperative efforts, and facilitates collaboration. In addition, the SDI movement also has two additional advantages. First, it allows a more effective and efficient interaction with geospatial data and, second, it helps to stimulate the market for the geospatial industry

(Bernard, 2002). The general approach to developing an SDI is to first understand how and where geospatial data is created. Most SDIs or geospatial clearinghouses base their first level data collection efforts on framework data (FGDC95). Framework data is created by government agencies—local, state, federal, or regional for the purpose of conducting their business such as development and maintenance of roads, levying taxes, monitoring streams, or creating land use ordinances. These business practices translate themselves, in the geopspatial data world, into transportation network data, parcel or cadastral data, water quality data, aerial photographs, or interpreted satellite imagery. Other organizations can then build upon this framework data to create watershed assessments, economic development plans, or biodiversity and habitat maps. This pyramid of data sharing—from local to national—has been the cornerstone of the original concept of the SDI and considered a fundamental key to building an SDI (Rajabifard & Williamson, 2001).

The SDI movement now encompasses countries and regions all over the world and is now considered a global movement and potential global resource. Many countries maintain now clearinghouses participating in regional efforts. One effort along these lines is the GSDI (Nebert, 2004). The GSDI, which resulted from meetings held in 1995, is a non-profit organization working to further the goals of data sharing and to bring attention to the value of the SDI movement with a particular emphasis on developing nations (Stevens et al., 2004). Other projects including the Geographic Information Network in Europe (GINIE) project are working toward collaboration and cooperation in sharing geospatial data (Craglia, 2003). As of 2006, there were approximately 500 geospatial data clearinghouses throughout the world. The activities of the clearinghouses range from coordinating data acquisition and developing data standards to developing applications and services for public use with an average operating cost of approximately € 1,500,000 per year (approximately $ 1,875,000) (Crompvoets et al., 2006).

EVOLUTION OF SERVICES AND ACCESS IN THE GEOSPATIAL DATA CLEARINGHOUSE

There are several developmental phases that geospatial data clearinghouses engage in to become fully operational and integrated into a larger SDI, e.g., data acquisition and documentation, data access and retrieval capabilities, storage architecture development, and application development. These phases can be sequential or can be performed simultaneously but all must be addressed. It is important to note that technology, both internal and external to the clearinghouse, changes rapidly and therefore any clearinghouse must be developed to be dynamic to meet the changing nature of the technology and the changing needs of its users. Each geospatial data clearinghouse also must address the particulars of their organization such as available software, hardware, database environment, technical capabilities of staff, and the requirements of their primary clients or users. In some cases, clearinghouses have undertaken an effort to develop user requirements and assess needs prior to implementation of new services or architectures. The user needs and requirements assessment addresses all phases of the clearinghouse from both internal and external perspectives and provides the framework with which to build services and organizational capability (Kelly & Stauffer, 2000). Within the requirements phase, examination of resources available to the clearinghouse must be determined and if inadequate, acquired. There is little doubt that the key to success relies heavily on the resources of the geospatial data clearinghouse and its ability to store and provide access to large datasets and thousands of data files (Kelly & Stauffer, 2000). Another equally significant component of building an SDI is identifying how the resource will support activities in the region. The clearinghouse can bring together disparate data sets, store data for those organizations that are unable to store or provide access to their own information, and can offer access to data that crosses boundaries or regions to enable efforts that are outside traditional jurisdictions of agencies or organizations (Rajabifard & Williamson, 2001).

Metadata

The key component to any geospatial data clearinghouse is geospatial metadata. The metadata forms the core of all other operations and should be addressed in the initial phase of clearinghouse development. The Federal Geographic Data Committee (FGDC) developed its initial standards for geospatial metadata in the mid 1990's. This standard, which is used as the basis for metadata in geospatial data clearinghouses today is re-

ferred to as the Content Standard for Digital Geospatial Metadata (CSDGM) (FGDC98). The impact of CSDGM cannot be overstated. The early metadata standard has grown over the years and been adapted and accepted internationally by the International Organization for Standardization (ISO). The metadata not only serves as a mechanism to document the data but also serves as the standard basis for distributed sharing across clearinghouses through centralized gateways or national SDI (Figure 1). The standards used to query remote catalogs have their origins in the development of the ANSI Z39.50 standard (now known as the ISO 23950 Search and Retrieval Protocol) which was originally designed for libraries to search and retrieve records from remote library catalogs (Nebert, 2004). In addition to addressing the metadata standard, a secondary yet equally important issue is format. Initially metadata was either HTML or text format and then parsed using a metadata parser into the standard fields. One of the first clearinghouses to implement XML (Extensible Markup Language) based metadata was Pennsylvania Spatial Data Access (PASDA). PASDA, which was first developed in 1996, utilized XML metadata in an effort to better manage data and to make updates and alterations to metadata more efficient (Kelly & Stauffer, 2000).

Data Acquisition

One of the most significant challenges for any geospatial data clearinghouse is the acquisition of data. Each country, state, and region face their own internal issues related to data sharing such as legal liability questions and right to know laws, therefore the data and services for each clearinghouse differ. The ideal concept for sharing would be from local government, since it can be the most detailed and up-to-date, to state and federal government clearinghouses—the hierarchy of data sharing (Figure 2). However, it can be difficult to acquire local government data unless partnerships are developed to encourage and enable sharing of local data (McDougall et al., 2005).

There are some success stories that do demonstrate the benefits of partnerships and some even with major metropolitan areas. The City of Philadelphia, which maintains one of the most advanced geospatial enter-

Figure 1. User access to remote data stores via SDI gateway

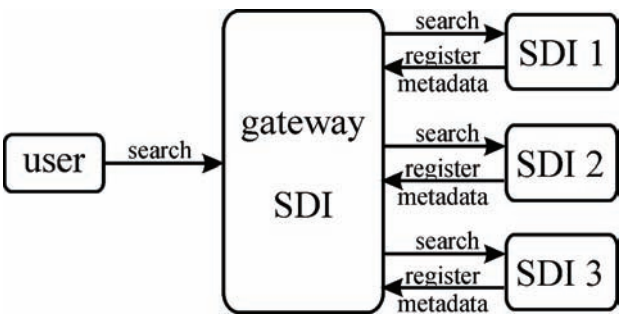


Figure 2. Traditional data sharing process from local governments to state or regional clearinghouses to national SDI gateway

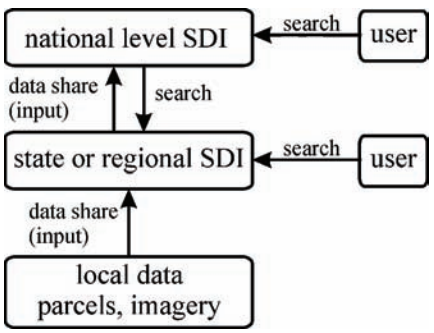


Figure 3. Internet Map Service approach to sharing data

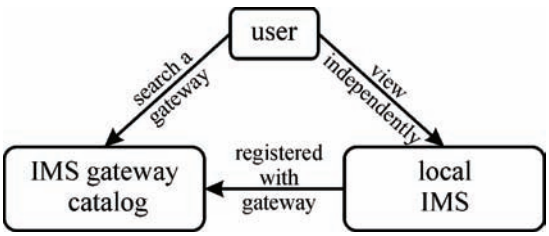
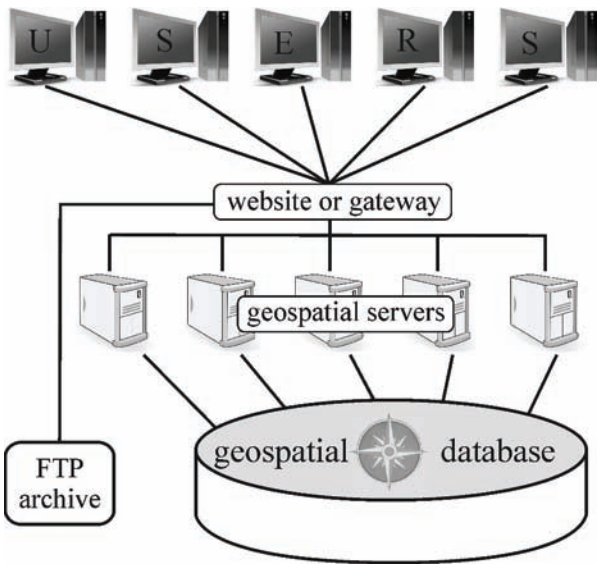


Figure 4. Common architecture of a current geospatial data clearinghouse



prise architectures in the US, shares its data through the state geospatial data clearinghouse PASDA.

However, in general the partnership approach has lead to spotty participation by local and regional organizations therefore many geospatial clearinghouses base their initial efforts on acquiring data which is freely available and which has no distribution restrictions. In the United States, this data tends to be from the Federal government. The initial data sets are comprised of elevation, scanned geo-referenced topographic maps, and aerial photographs. If the geospatial data clearinghouse is state-related, additional more detailed information such as roads, streams, boundary files, parks and recreational data are added to the data collection.

However, changes in technology have altered the concept of how data is shared. The increase in web-based GIS applications or Internet Map Services (IMS) has allowed local governments, as well as others, to share their data via IMS catalogs by registering their "service" versus sharing their data files. In this approach, access is still provided to the data, although with limited download capabilities, and local control and hosting is maintained (Figure 3).

Architecture

Geospatial data have traditionally been accessed via the Internet through file transfer protocol (FTP) or more recently viewed through web based interfaces or IMS. The earliest geospatial data clearinghouses required an FTP server, metadata that could be in HTML (Hypertext Markup Language), later XML, and simple search mechanisms driven by keywords or descriptors in the metadata. In cases such as these the onus was on the user or client to access and manipulate the data into their own environment. However, recent developments in database management and software, and advances in the overall technologies related to the Internet are

driving the rapid changes in geospatial clearinghouses. Users who are requiring more web-based services, customizing capabilities, and faster performance times combined with the exponential growth in available data has increased the requirements for clearinghouses (Kelly & Stauffer, 2000). Within the past six years, user needs have moved clearinghouse operations from simple FTP sites to complex organizations supporting large relational database management systems (RDBMS) composed of vast vector and raster data stores, specialized customization engines—reprojectors and data clipping utilities, access to temporal data, analysis tools and multiple IMS applications while still maintaining archival copies and direct FTP download capabilities (Figure 4).

EMERGING TRENDS IN GEOSPATIAL DATA CLEARINGHOSUE DEVELOPMENT

The dramatic shift from serving data and metadata through a simple web based interface to storing and providing access to thousands of unique data sets through multiple applications has placed the clearinghouse movement on the cutting edge of geospatial technologies. These cutting technologies stem from advancements in managing data in a Relational Database Management system (RDBMS) with the accompanying increase in performance of IMS and temporal data integration techniques.

RDBMS and Its Impact on Data Management

The current movement in geospatial data clearinghouses is to support and manage data within an RDBMS. In essence this approach has altered not only how the data is managed but has also improved the performance and utility of the data by enabling application developers to create IMS and other services that mirror desktop capabilities. Advancements in database technology and software such as ESRI Spatial Database Engine (SDE), in combination with databases such as Oracle or DB2, can support the management of large quantities of vector and raster data. The data is stored within the database in table spaces that are referenced, within the DB2 architecture, by configuration keywords that point to attribute table spaces and coordinate table spaces. For vector data, a spatial index grid is created which allows features to be indexed by one or more grids and stored in the database based on spatial index keys that correspond to the index grid. In general, raster data, which are in essence multidimensional grids can be stored as binary large objects or BLOBs within the table spaces. Interfacing with the database is a broker such as SDE which enables both input and retrieval of the raster BLOBs. In addition, techniques, such as the use of pyramids that render the image at a reduced resolution, increase the performance of the database for the end user. In addition, a clearinghouse that uses a RDBMS architecture is more readily able to manage temporal or dynamic data and automate processes for updating applications and services (Kelly et al, 2007). These advances in managing and retrieving data within the RDBMS/SDE environment have substantially increased performance and speed even more so in an IMS environment (Chaowei, et al, 2005). Another impact of this trend is that as relational database management become mainstreamed with the more user friendly and affordable databases such as MySQL, an open source RDBMS that uses Structured Query Language (SQL), the ability for smaller organizations to develop higher functioning IMS is within the realm of possibility.

Internet Map Services

In the past few years, IMS, which encompass everything from stand alone applications dedicated to a specific theme or dataset (i.e., "My Watershed Mapper"), have grown to represent the single most important trend in the geospatial data clearinghouse movement. The early period of IMS development included applications comprised of a set of data, predefined by the developer, which users could view via an interactive map interface within an HTML page. However, there were few if any customization capabilities and limited download capabilities. The user was simply viewing the data and turning data layers on and off. Another component of early IMS development was the use of a Web GIS interface or map as part of a search engine to access data and metadata from data clearinghouses or spatial data distribution websites (Kraak, 2004). Building on the advances of RDBMS and software, IMS developers have been able to set aside the historical constraints that hindered development in the 1990s. Initially, data used for IMS were stored in a flat

file structure, such as ESRI shapefiles with which the Internet mapping server interacted directly. While this was somewhat effective for small files in the kilobyte or single megabyte range, it became more cumbersome and inefficient for the increasing amount and size of detailed vector and raster data, such as high-resolution aerial photography. But as the use of RDBMS became the backbone of many geospatial data clearinghouse architectures, IMS developers began to take advantage of the data retrieval performance and efficiency of relational databases.

Changes in IMS have emerged over the past few years that herald a change in how users interact with geospatial data clearinghouses. The development of Web Feature Services (WFS) and Web Map Services (WMS) are pushing the envelope of an open GIS environment and enabling interoperability across platforms. There are several types of emerging IMS. These include the feature service and image service. The image service allows the user to view a snapshot of the data. This type of service is particularly meaningful for those utilizing raster data, aerial photography, or other static data sets. The feature service is the more intelligent of the two as it brings with it the spatial features and geometry of the data and allows the user to determine the functionality and components such as the symbology of the data. WFS is particularly applicable for use with real-time data streaming (Zhang & Li, 2005).

Changes in technology have allowed geospatial data clearinghouses to deploy applications and services containing terabytes of spatial data within acceptable time frames and with improved performance. IMS has moved the geospatial data clearinghouse from a provider of data to download to interactive, user centered resources that allow users to virtually bring terabytes of data to their desktop without downloading a single file.

Temporal Data

As geospatial data clearinghouses have crossed over into the IMS environment, the issue of integrating temporal data has become an emerging challenge (Kelly et al, 2007). The expectations for clearinghouses are moving toward not only providing terabytes of data but also toward providing data that is dynamic. Temporal data, unlike traditional spatial data in which the attributes remain constant, has constantly changing attributes and numerous data formats and types with which to contend (Van der Wel, et al., 2004). Temporal data by nature presents numerous challenges to the geospatial data clearinghouse because it carries with it the added dimension of time. An prime example of the complexity of temporal data integration is weather data. The development of services based on temporal information such as weather data must be undertaken with the understanding that this data is unique in format and that the frequency of updates require that any service be refreshed "on the fly" so the information will always be up to date. The number of surface points and types of data such as data taken from satellite or radar can overwhelm even a sophisticated server architecture (Liknes, 2000). However, the significance of these data to users from emergency management communities to health and welfare agencies cannot be underestimated. Therefore, it is imperative that geospatial data clearinghouses play a role in providing access to this vital data.

CONCLUSION

The changes in geospatial data clearinghouse structure and services in less than a decade have been dramatic and have had a significant impact on the financial and technical requirements for supporting an SDI geospatial data clearinghouse. As stated earlier in this section, the average cost of maintaining and expanding a clearinghouse is clear since most of the costs are assessed on personnel, hardware, software, and other apparent expenses; it is more difficult to assess the financial benefit (Gillespie, 2000). In addition, despite many changes in technology and the increasing amount of accessible data, some challenges remain. Local data is still underrepresented in the clearinghouse environment and the ever-changing technology landscape requires that geospatial data clearinghouse operations be dynamic and flexible. As trends such as IMS, the use of temporal data, and the enhancement of relational database architectures continue, geospatial data clearinghouses will be faced with providing growing amounts of data to ever more savvy and knowledgeable users.

REFERENCES

Bernard, L. (2002, April). Experiences from an implementation Testbed to set up a national SDI. Paper

presented at the *5th AGILE Conference on Geographic Information Science, Palma, Spain.*

Chaowei, P.Y., Wong, D., Ruixin, Y., Menas, K., & Qi, L.(2005). Performance-improving techniques in web-based GIS. *International Journal of Geographical Information Science*, 19 (3), 319-342.

Clinton, W.J. (1994). Coordinating geographic data acquisition and access to the National Geospatial Data Infrastructure. Executive Order 12096, *Federal Register*, 17671-4. Washington: D.C.

Craglia, M., Annoni, A., Klopfer, M., Corbin, C., Hecht, L., Pichler, G., & Smits, P. (Eds.) (2003). Geographic information in wider Europe, geographic information network in Europe (GINIE), http://www.gis.org/ginie/doc/GINIE_finalreport.pdf.

Crompvoets, J., Bregt, A., Rajabifard, A., Williamson, I. (2004). Assessing the worldwide developments of national spatial data clearinghouses. *International Journal of Geographical Information Science,* 18(7), 655-689.

FGDC95: Federal Geographic Data Committee, (1995). *Development of a national digital geospatial data framework.* Washington, D.C.

FGDC98: Federal Geographic Data Committee (1998). *Content standard for digital geospatial metadata.* Washington, D.C.

Gillespie, S. (2000). An empirical approach to estimating GIS benefits. *URISA Journal*, 12 (1), 7-14.

Kelly, M.C. & Stauffer, B.E. (2000). *User needs and operations requirements for the Pennsylvania Geospatial Data Clearinghouse.* University Park, Pennsylvania: The Pennsylvania State University, Environmental Resources Research Institute.

Kelly, M. C., Haupt, B.J., & Baxter, R. E. (2007). The evolution of spatial data infrastructure (SDI) geospatial data clearinghouses: Architecture and services. In *Encyclopedia of database technologies and applications* (invited, accepted). Hershey, Pennsylvania, USA: Idea Group, Inc.

Kraak, M.J. (2004). The role of the map in a Web-GIS environment. *Journal of Geographical Systems,* 6(2), 83-93.

McDougall, K., Rajabifard, A., & Williamson, I. (2005, September). What will motivate local governments to share spatial information? Paper presented at the *National Biennial Conference of the Spatial Sciences Institute,* Melbourne, AU.

National Research Council, Mapping Science Committee. (1993). *Toward a coordinated spatial data infrastructure for the nation*. Washington, D.C.: National Academy Press.

Nebert, D. (Ed.) (2004). *Developing spatial data infrastructures: The SDI cookbook* (v.2). Global Spatial Data Infrastructure Association. http://www.gsdi.org.

Rajabifard, A., & Williamson, I. (2001). Spatial data infrastructures: concept, SDI hierarchy, and future directions. Paper presented at the *Geomatics 80 Conference*, Tehran, Iran.

Stevens, A.R., Thackrey, K., Lance, K. (2004, February). Global spatial data infrastructure (GSDI): Finding and providing tools to facilitate capacity building. Paper presented at the *7th Global Spatial Data Infrastructure conference,* Bangalore, India.

Van der Wel, F., Peridigao, A., Pawel, M. , Barszczynska, M. , & Kubacka, D. (2004): COST 719: Interoperability and integration issues of GIS data in climatology and meteorology. *Proceedings of the 10th EC GI & GIS Workshop*, ESDI State of the Art, June 23-25 2004, Warsaw, Poland.

KEY TERMS

BLOB: Binary Large Object is a collection of binary data stored as a single entity in a database management system

Feature Service: The OpenGIS Web Feature Service Interface Standard (WFS) is an interface allowing requests for geographical features across the web using platform-independent calls. The XML-based GML is the default payload encoding for transporting the geographic features.

ISO 23950 and Z39.50: International standard specifying a client/server based protocol for information retrieval from remote networks or databases.

Metadata: Metadata (Greek meta "after" and Latin data "information") are data that describe other data. Generally, a set of metadata describes a single set of data, called a resource. Metadata is machine understandable information for the web.

Open GIS: Open GIS is the full integration of geospatial data into mainstream information technology. What this means is that GIS users would be able to freely exchange data over a range of GIS software systems and networks without having to worry about format conversion or proprietary data types.

SDE: SDE (Spatial Data Engine) is a software product from Environmental Systems Research Institute (ESRI) that stores GIS data in a relational database, such as Oracle or Informix. SDE manages the data inside of tables in the database, and handles data input and retrieval.

Web Mapping Services (WMS): A Web Map Service (WMS) produces maps of geo-referenced data and images (e.g. GIF, JPG) of geospatial data.

XML : Extensible Markup Language (XML) was created by W3C to describe data, specify document structure (similar to HTML), and to assist in transfer of data.

Evolutionary Approach to Dimensionality Reduction

Amit Saxena
Guru Ghasida University, Bilaspur, India

Megha Kothari
St. Peter's University, Chennai, India

Navneet Pandey
Indian Institute of Technology, Delhi, India

INTRODUCTION

Excess of data due to different voluminous storage and online devices has become a bottleneck to seek meaningful information therein and we are information wise rich but knowledge wise poor. One of the major problems in extracting knowledge from large databases is the size of dimension i.e. number of features, of databases. More often than not, it is observed that some features do not affect the performance of a classifier. There could be features that are derogatory in nature and degrade the performance of classifiers used subsequently for dimensionality reduction (DR). Thus one can have redundant features, bad features and highly correlated features. Removing such features not only improves the performance of the system but also makes the learning task much simpler. Data mining as a multidisciplinary joint effort from databases, machine learning, and statistics, is championing in turning mountains of data into nuggets (Mitra, Murthy, & Pal, 2002)

Feature Analysis

DR is achieved through feature analysis which includes feature selection (FS) and feature extraction (FE). The term FS refers to selecting the best subset of the input feature set whereas creating new features based on transformation or combination of the original feature set is called FE. FS and FE can be achieved using supervised and unsupervised approaches. In a supervised approach, class label of each data pattern is given and the process of selection will use this knowledge for determining the accuracy of classification whereas in unsupervised FS, class level is not given and process will apply natural clustering of the data sets.

BACKGROUND

Feature Selection (FS)

The main task of FS is to select the most discriminatory features from original feature set to lower the dimension of pattern space in terms of internal information of feature samples. Ho (Ho, 1998) combined and constructed multiple classifiers using randomly selected features which can achieve better performance in classification than using the complete set of features. The only way to guarantee the selection of an optimal feature vector is an exhaustive search of all possible subset of features (Zhang, Verma, & Kumar, 2005).

Feature Selection Methods

In the FS procedures, four basic stages are distinguished:

1 **Generation procedure:** In this stage a possible subset of features to represent the problem is determined. This procedure is carried according to one of the standard methods used for this purpose.
2. **Evaluation function:** In this stage the subset of features selected in the previous stage is evaluated according to some fitness function.
3. **Stopping criterion:** It is verified if the evaluation of the selected subset satisfies the stopping criterion defined for the searching procedure.

4. **Validation procedure:** It will be used to test the quality of the selected subset of features.

FS methods can be divided into two categories: the wrapper method and the filter one. In the wrapper methods, the classification accuracy is employed to evaluate feature subsets, whereas in the filter methods, various measurements may be used as FS criteria. The wrapper methods may perform better, but huge computational effort is required (Chow, & Huang, 2005). Thus, it is difficult to deal with large feature sets, such as the gene (feature) set of a cDNA data. A hybrid method suggested by Liu (Liu, & Yu 2005) attempts to take advantage of both the methods by exploiting their different evaluation criteria in different search stages. The FS criteria in the filter methods fall into two categories: the classifier-parameter-based criterion and the classifier-free one.

An unsupervised algorithm (Mitra, Murthy, & Pal, 2002) uses feature dependency/similarity for redundancy reduction. The method involves partitioning of the original feature set into some distinct subsets or clusters so that the features within a cluster are highly similar while those in different clusters are dissimilar. A single feature from each such cluster is then selected to constitute the resulting reduced subset.

Use of soft computing methods like GA, fuzzy logic and Neural Networks for FS and Feature ranking is suggested in (Pal, 1999). FS method can be categorized (Muni, Pal & Das, 2006) into five groups based on the evaluation function, distance, information, dependence, consistency (all filter types) and classifier error rate (wrapper type). It is stated in (Setiono, & Liu, 1997) that the process of FS works opposite to ID3 (Quinlan,1993). Instead of selecting one attribute at a time, it starts with taking whole set of attributes and removes irrelevant attribute one by one using a three layer feed forward neural network. In (Basak, Pal, 2000) neuro-fuzzy approach was used for unsupervised FS and comparison is done with other supervised approaches. Recently developed Best Incremental Ranked Subset (BIRS) based algorithm (Roberto Ruiza, José C. Riquelmea, Jesús S. Aguilar-Ruizb, 2006) presents a fast search through the attribute space and any classifier can be embedded into it as evaluator. BIRS chooses a small subset of genes from the original set (0.0018% on average) with similar predictive performance to others. For very high dimensional datasets, wrapper-based methods might be computationally unfeasible, so BIRS turns out a fast technique that provides good performance in prediction accuracy.

Feature Extraction (FE)

In a supervised approach, FE is performed by a technique called discriminant analysis (Steve De Backer, 2002). Another supervised criterion for non-Gaussian class-conditional densities is based on the Patrick-Fisher distance using Parzen density estimates .The best known unsupervised feature extractor is the principal component analysis (PCA) or Karhunen-Lo`eve expansion, that computes the $\boldsymbol{d}$ largest eigenvectors from $\mathbf{D} \times \mathbf{D}$ covariance matrix of the $\boldsymbol{n}$, **D**-dimensional patterns. Since PCA uses the most expressive features (eigenvectors with largest eigenvalues), it effectively approximates the data by a linear subspace using the mean squared error criterion.

MAIN FOCUS

Evolutionary Approach for DR

Most of the researchers prefer to apply evolutionary approach to select features. There are evolutionary programming (EP) approaches like particle swarm optimization (PSO), ant colony optimization (PSO), genetic algorithms (GA); however GA is the most widely used approach whereas ACO and PSO are the emerging area in DR. Before describing GA approaches for DR, we outline PSO and ACO.

Particle Swarm Optimization (PSO)

Particle swarm optimizers are population-based optimization algorithms modeled after the simulation of social behavior of bird flocks (Kennedy, Eberhart, 2001). A swarm of individuals, called particles, fly through the search space. Each particle represents a candidate solution to the optimization problem. The position of a particle is influenced by the best position visited by itself (i.e. its own experience) and the position of the best particle in its neighborhood (i.e. the experience of neighboring particles called *gbest*/*lbest*) (Engelbrecht, 2002).

Ant Colony Optimization (ACO)

Dorigo (Dorigo, & Stützle, 2004) introduces the behavior of ants. It is a probabilistic technique to solve computational problems. When ants walk randomly in search of food, after finding food, they lay down a substance called pheromone in their trails. Other ants follow this trail laying down more and more pheromone to strengthen the path.

Genetic Algorithms (GA) Based FS

The GA is biologically inspired and has many mechanisms mimicking natural evolution (Oh, Lee , & Moon 2004). GA is naturally applicable to FS since the problem has an exponential search space. The pioneering work by Siedlecki (Siedlecki, Sklansky, 1989) demonstrated evidence for the superiority of the GA compared to representative classical algorithms.

Genetic algorithm (GA) (Goldberg, 1989) is one of the powerful evolutionary algorithms used for simultaneous FS. In a typical GA, each chromosome represents a prospective solution of the problem. The set of all chromosomes is usually called a population. The chromosome can either be represented as a binary string or in real code depending on the nature of a problem. The problem is associated with a fitness function. The population goes through a repeated set of iterations with crossover and mutation operation to find a better solution (better fitness value) after each iteration. At a certain fitness level or after a certain number of iterations, the procedure is stopped and the chromosome giving the best fitness value is preserved as the best solution of the problem.

Fitness function for DR

Each chromosome in the population is first made to represent all features of the pattern in a binary manner.

Sammon's Error used as the fitness function is given as follows.

Let $\boldsymbol{X = \{x_k \mid x_k = (x_{k1}, x_{k2}, \ldots, x_{kp})^T,\ k=1,2,\ldots,n\}}$ be the set of $\boldsymbol{n}$ input vectors and let $\boldsymbol{Y = \{y_k \mid y_k = (y_{k1}, y_{k2}, \ldots, y_{kq})^T,\ k=1,2,\ldots,n\}}$ be the unknown vectors to be found. Let $\mathbf{d^*_{ij} = d(x_i, x_j)}$, $\mathbf{x_i, x_j} \in \boldsymbol{X}$ and $\boldsymbol{d_{ij} = d(y_i, y_j)}$, $\mathbf{y_i, y_j} \in \boldsymbol{Y}$, where $\boldsymbol{d(x_i, x_j)}$ be the Euclidean distance between $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$. Sammon Error $\boldsymbol{E}$ is given by,

$$E = \frac{1}{\sum_{i<j} d^*_{ij}} \sum_{i<j} \frac{(d^*_{ij} - d_{ij})^2}{d^*_{ij}} \qquad (1)$$

Procedure

The entire procedure is outlined below.

A. Initialize the population

Let p be the length of each chromosome which is equal to the number of features present in the original data set. Let N_p be the size of the population matrix having N number of chromosomes. Initialize each chromosome of the population by randomly assigning 0 and 1 to its different positions.

B. Measure of fitness

If the i^{th} bit of the chromosome is a 1, then i^{th} feature of the original data set is allowed to participate while computing Sammon Error. Compute d^*_{ij} and d_{ij} for the data set for reduced and entire data set respectively. Calculate the fitness of each chromosome by computing the Sammon Error E given by Eq.(1).

C. Selection, crossover and mutation

Using roulette wheel selection method, (or any other selection criterion like uniform random) different pairs of chromosomes are selected for cross over process. The selected pairs are allowed to undergo a crossover (here also one can apply a single point or a two point crossover) operation to generate new offspring. These offspring are again selected randomly for mutation. In this manner a new population is constructed.

D. Termination of GA

Compute the fitness of each chromosome of new population and compare the fitness of each chromosome of new population with that in the initial one. Retain the chromosome for which Sammon Error is lesser in the initial population than that in the new population otherwise replace the chromosome of new population with that in the initial population. Stop the GA after a sufficiently large number of iterations. Note the number of the features and their locations in the final chromosome having least Sammon Error.

E. Validation

Apply k-means clustering (for unsupervised classification) and k-NN classifier (k-Nearest Neighbor, for supervised classification) techniques to compute and compare the accuracies of the reduced and entire data sets.

After, a reduced data set is obtained, it will be appropriate to ensure that the topology *i.e.* the structure of the original data set is preserved in the reduced set. For this purpose, we compute the proximity matrices of the original and reduced data sets and compute correlation coefficients of the two matrices. High values of correlation coefficients confirm that the structure of the data set is preserved in reduced data set (Saxena, Kothari, 2007). A fuzzy approach is used before for DR with structure preserving (Pal, Mandal, 2002) but GA based approaches have not been reported so far.

GAs have been used to search for feature subsets in conjunction with several classification methods (Cantú-Paz, 2004) such as neural networks, decision trees, k-nearest neighbors rules and Naive Bayes.

The GA for FS can be applied with some penalty (Oh, Lee, & Moon, 2004). In order to force a feature subset to satisfy the given subset size requirement, the size value, *d* is taken as a constraint and a penalty is imposed on chromosomes breaking this constraint. A novel hybrid GA (HGA) is proposed to solve the FS problem. Local search operations parameterized with ripple factors were devised and embedded in the HGAs. Using standard data sets, experiments justified the proposed method.

The GA Feature Extractor

Here, the objective function consists of the classification performance obtained by a k-NN classifier, using the value of features provided by the GA (Raymer, 2000). In order to prevent any initial bias due to the natural ranges of values for the original features, input feature values are normalized over the range [1.0, 10.0]. Prior to each GA experiment, the k-NN classifier is trained with a set of patterns with an equal number of samples of each class. A second set of patterns, disjoint to the training set, is set aside to serve as a tuning set. To evaluate a particular set of weights, the GA scales the training and tuning set feature values according to the weight set, and classified the tuning set with the weighted k-NN classifier. The performance of the weight set is determined according to the classification accuracy on the tuning set, the balance in accuracy among classes, the number of correct votes cast during k-NN classification, and the parsimony of the weight set (the number of features eliminated from classification by masking). Since the particular GA engine used minimizes the objective function, the following error function is used to evaluate each weight w^*

$$\text{error}(w^*) = C_{pred}(\text{ no. of incorrect predictions}) + C_{rmask}(\textit{ no. of unmasked features}) + C_{votes}(\textit{ no. of incorrect votes}) + C_{diff}(\textit{ difference in accuracy between classes})$$

where C_{pred}, C_{rmask}, C_{votes}, C_{diff} are anti-fitness function coefficients.

In most cases, GAs, combine different optimization objectives into a single objective function. The main drawback of this kind of strategy lies in the difficulty of exploring different possibilities of trade-offs among objectives. Morita (Morita, Sabourin, Bortolozzi, & Suen 2003) proposed a methodology for FS in unsupervised learning for handwritten month word recognition. It makes use of the Nondominated Sorting genetic algorithm (NSGA) proposed by Srinivas (Srinivas, Deb, 1995) which deals with multi-objective optimization. The objective is to find a set of nondominant solutions which contain the more discriminant features and the more pertinent number of clusters. Two criteria are used to guide the search: minimization of the number of features and minimization of a validity index that measures the quality of clusters. A standard K-means algorithm is applied to form the given number of clusters based on the selected features.

In another approach, using a new method called ReliefF (Zhang, Wang, Zhao, & Yang, 2003), the key idea is to estimate the quality of attributes according to how well their values distinguish between the instances that are near to each other. For that purpose, given a randomly selected instance *R*, ReliefF searches for k-NN from the same class, called nearHits, and also k-NN from each of the different classes, called nearMisses. The quality estimation w [A] for each A is updated depending on *R*, nearHits and nearMisses. In the update formula, the contribution of all the hits and misses are averaged. The process is repeated for m times to R retum weights of all features, where m is a user-defined parameter. ReliefF is fast, not lim-

ited by data types, fairly attribute noise-tolerant, and unaffected by feature interaction, but it does not deal well with redundant features. GA-Wrapper and ReliefF-GA-Wrapper methods are proposed and shown experimentally that ReliefF-GA-Wrapper gets better accuracy than GA-Wrapper.

An online FS algorithm using genetic programming (GP), named as GPmfts (multitree genetic programming based FS), is presented in (Muni, Pal, & Das, 2006). For a *c*–class problem, a population of classifiers, each having c trees is constructed using a randomly chosen feature subset. The size of the feature subset is determined probabilistically by assigning higher probabilities to smaller sizes. The classifiers which are more accurate using a small number of features are given higher probability to pass through the GP evolutionary operations. The method automatically selects the required features while designing classifiers for a multi-category classification problem.

FUTURE TRENDS

DR is an important component of data mining. Besides GA, there are several EP based DR techniques like PSO, ACO which are still unexplored and it will be interesting to apply these techniques. In addition, other techniques like wavelet, PCA , decision trees, rough sets can also be applied for DR. A good hybridization of existing techniques can also be modeled to take merits of inherent individual techniques.

CONCLUSION

This article presents various DR techniques using EP approaches in general and GA in particular. It can be observed that every technique has its own merits and limitations depending on the size and nature of the data sets. A Data set for which FS and FE require exhaustive search, evolutionary/GA methods is a natural selection as in the case of handwritten digit string recognition (Oliveira, & Sabourin, 2003) because GA makes efforts to simultaneous allocation of search in many regions of the search space, this is also known as implicit parallelism. It can be noted from various results presented by researchers that GA with variations in crossover, selection, mutation, and even in fitness function can be made more effective in finding an optimal subset of features.

REFERENCES

Beltrán N. H., Duarte. M. M. A., Salah S. A., Bustos M. A., Peña N. A. I., Loyola E. A., & Galocha J. W. (2005). Feature selection algorithms using Chilean wine chromatograms as examples. *Journal of Food Engineering, 67*(4), 483-490.

Chow T. W. S., & Huang D. (2005, Janunary). Estimating optimal feature subsets using efficient estimation of high-dimensional mutual information. *IEEE Transactions on Neural Networks, 16*(1), 213-224.

Dorigo M., & Stützle T. (2004). Ant colony optimization. *MIT Press*. (ISBN 0-262-04219-3).

Engelbrecht A. (2002). Computational Intelligence- An Introduction. *John Wiley and Sons*.

Goldberg D. (1989). Genetic algorithms in search, optimization, and machine learning. *Reading, MA: Addison-Wesley*.

Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis. And Machine Intelligence 20*(8), 832–844.

Basak J., De R. K., & Pal S. K. (2000, March) Unsupervised feature selection using a neuro-fuzzy approach. *IEEE Transactions on Neural Networks, 11*(2), 366-375.

Kennedy., & Eberhart R. (2001). Swarm Intelligence. *Morgan Kaufmann*.

Liu H., & Yu L. (2005, April). Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering, 17*(4), 491-502.

Mitra P., Murthy C. A., & Pal S. K. (2002). Unsupervised feature selection using feature similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 24(3), 301-312.

Morita M., Sabourin R., Bortolozzi F., & Suen C. Y. (2003). Unsupervised feature selection using multi-objective genetic algorithms for handwritten word recognition. In *Proceedings of the Seventh International*

Conference on Document Analysis and Recognition (ICDAR'03) .

Muni D. P., Pal N. R., & Das J. (2006, February). Genetic programming for simultaneous feature selection and classifier design. *IEEE Transactions on Systems*, Man, and Cybernetics—PART B,36(1).

Oh I.S., Lee J.S., & Moon B.R. (2004). Hybrid genetic algorithms for feature selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11), 1424-1437.

Cantú-Paz E. (2004). Feature subset selection, class separability, and genetic algorithms. *Lecture Notes in Computer Science* 3102, 959-970.

Oh II-Seok, Lee J. S., & Moon B. R. (2004, November). Hybrid genetic algorithms for feature selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 26*(11).

Oliveria L. S., & Sabourin R. (2003). A methodology for feature selection using Multi-objective genetic algorithms for handwritten digit string recognition. *International Journal of Pattern Recognition and Artificial Intelligence, 17*(6), 903-929.

Pal N. R. (1999). Soft computing for feature analysis. *Fuzzy Sets and Systems*, *103*(2), 201-221.

Pal N. R., Eluri V. K., & Mandal G. K. (2002, June). Fuzzy logic approaches to structure preserving dimensionality reduction. *IEEE Transactions on Fuzzy Systems*, *10*(3), 277-286.

Quinlan J. R. (1993). C4.5: *Programs for machine learning*. San Mateo, CA: Morgan Kaufmann.

Raymer M. L., Punch W. F., Goodman E. D., Kuhn L. A., & Jain A. K. (2000, July). Dimensionality reduction using genetic algorithms. *IEEE Transactions on Evolutionary Computation*, *4*(2), 164-171.

Roberto Ruiz, José C. Riquelme, Jesús S. Aguilar-Ruiz (2006). Incremental wrapper- based gene selection from microarray data for cancer classification. *Pattern Recognition, 39*(2006), 2383 – 2392.

Saxena Amit, & Kothari Megha (2007). Unsupervised approach for structure preserving dimensionality reduction. In *Proceedings of ICAPR07 (The 6th International Conference on Advances in Pattern Recognition 2007) pp. 315-318, ISBN – 13 978-981-270-553-2, 10 981-270-553-8*. World Scientific Publishing Co. Pte. Ltd. Singapore.

Setiono R., & Liu H. (1997, May). Neural-network feature selector. *IEEE Transactions on Neural Networks*, *8*(3), 654-662.

Siedlecki W., & Sklansky J. (1989). A note on genetic algorithms for large-scale feature selection. *Pattern Recognition Letters*, *10*(5), 335–347.

Srinivas N., & Deb K. (1995). Multi-objective optimization using non-dominated sorting in genetic algorithms. *IEEE Transactions on Evolutionary Computation*, *2*(3), 221–248.

Steve D. B. (2002). *Unsupervised pattern recognition, dimensionality reduction and classification*. Ph.D. Dissertation. University of Antwerp.

Zhang P., Verma B., & Kumar K. (2005, May). Neural vs. Statistical classifier in conjunction with genetic algorithm based feature selection. *Pattern Recognition Letters*, *26*(7), 909-919.

Zhang L. X., Wang J. X., Zhao Y. N., & Yang Z. H. (2003). A novel hybrid feature selection algorithm: using relief estimation for GA-wrapper search. In *Proceedings of the Second International Conference on Machine Learning and Cybernetics*, *1*(1), 380-384.

KEY TERMS

Classification: It refers to labeling a class to a pattern in a data set. It can be supervised, when label is given and unsupervised when no labeling is given in advance. The accuracy of classification is a major criterion to achieve DR as removing features should not affect the classification accuracy of the data set.

Dimensionality Reduction (DR): The attributes in a data set are called features. The number of features in a data set is called its dimensionality. Eliminating redundant or harmful features from a data set is called DR.

Feature Analysis: It includes feature selection (FS) and feature extraction (FE). FS selects subset of features from the data set where as FE combines subsets of features to generate a new feature.

Genetic Algorithms (GA): These are randomized search and optimization methods based on principles of evolution and natural genetics. The algorithms apply crossover, selection, mutation and elitism operations to seek better solution with every execution.

Evolutionary Computation and Genetic Algorithms

William H. Hsu
Kansas State University, USA

INTRODUCTION

A **genetic algorithm** (**GA**) is a method used to find approximate solutions to difficult search, optimization, and machine learning problems (Goldberg, 1989) by applying principles of evolutionary biology to computer science. Genetic algorithms use biologically-derived techniques such as inheritance, mutation, natural selection, and recombination. They are a particular class of evolutionary algorithms.

Genetic algorithms are typically implemented as a computer simulation in which a population of abstract representations (called *chromosomes*) of candidate solutions (called *individuals*) to an optimization problem evolves toward better solutions. Traditionally, solutions are represented in binary as strings of 0s and 1s, but different encodings are also possible. The evolution starts from a population of completely random individuals and happens in generations. In each generation, multiple individuals are stochastically selected from the current population, modified (mutated or recombined) to form a new population, which becomes current in the next iteration of the algorithm.

BACKGROUND

Operation of a GA

The problem to be solved is represented by a list of parameters, referred to as *chromosomes* or genomes by analogy with genome biology. These parameters are used to drive an evaluation procedure. Chromosomes are typically represented as simple strings of data and instructions, in a manner not unlike instructions for a von Neumann machine. A wide variety of other data structures for storing chromosomes have also been tested, with varying degrees of success in different problem domains.

Initially several such parameter lists or chromosomes are generated. This may be totally random, or the programmer may seed the gene pool with "hints" to form an initial pool of possible solutions. This is called the *first generation pool*.

During each successive generation, each organism is evaluated, and a measure of quality or *fitness* is returned by a fitness function. The pool is sorted, with those having better fitness (representing better solutions to the problem) ranked at the top. "Better" in this context is relative, as initial solutions are all likely to be rather poor.

The next step is to generate a second generation pool of organisms, which is done using any or all of the genetic operators: selection, crossover (or recombination), and mutation. A pair of organisms is selected for breeding. Selection is biased towards elements of the initial generation which have better fitness, though it is usually not so biased that poorer elements have no chance to participate, in order to prevent the solution set from converging too early to a sub-optimal or local solution. There are several well-defined organism selection methods; roulette wheel selection and tournament selection are popular methods.

Following selection, the *crossover* (or *recombination*) operation is performed upon the selected chromosomes. Most genetic algorithms will have a single tweakable *probability of crossover* (P_c), typically between 0.6 and 1.0, which encodes the probability that two selected organisms will actually breed. A random number between 0 and 1 is generated, and if it falls under the crossover threshold, the organisms are mated; otherwise, they are propagated into the next generation unchanged. Crossover results in two new child chromosomes, which are added to the second generation pool. The chromosomes of the parents are mixed in some way during crossover, typically by simply swapping a portion of the underlying data structure (although other, more complex merging mechanisms have proved useful for certain types of problems.) This process is repeated with different parent organisms until there are an appropriate number of candidate solutions in the second generation pool.

The next step is to mutate the newly created offspring. Typical genetic algorithms have a fixed, very small *probability of mutation (P_m)* of perhaps 0.01 or less. A random number between 0 and 1 is generated; if it falls within the P_m range, the new child organism's chromosome is randomly mutated in some way, typically by simply randomly altering bits in the chromosome data structure.

These processes ultimately result in a second generation pool of chromosomes that is different from the initial generation. Generally the average degree of fitness will have increased by this procedure for the second generation pool, since only the best organisms from the first generation are selected for breeding. The entire process is repeated for this second generation: each organism in the second generation pool is then evaluated, the fitness value for each organism is obtained, pairs are selected for breeding, a third generation pool is generated, etc. The process is repeated until an organism is produced which gives a solution that is "good enough".

A slight variant of this method of pool generation is to allow some of the better organisms from the first generation to carry over to the second, unaltered. This form of genetic algorithm is known as an *elite selection strategy*.

MAIN THRUST OF THE CHAPTER

Observations

There are several general observations about the generation of solutions via a genetic algorithm:

- GAs may have a tendency to converge towards local solutions rather than global solutions to the problem to be solved.
- Operating on dynamic data sets is difficult, as genomes begin to converge early on towards solutions which may no longer be valid for later data. Several methods have been proposed to remedy this by increasing genetic diversity somehow and preventing early convergence, either by increasing the probability of mutation when the solution quality drops (called *triggered hypermutation*), or by occasionally introducing entirely new, randomly generated elements into the gene pool (called *random immigrants*).
- As time goes on, each generation will tend to have multiple copies of successful parameter lists, which require evaluation, and this can slow down processing.
- Selection is clearly an important genetic operator, but opinion is divided over the importance of crossover verses mutation. Some argue that crossover is the most important, while mutation is only necessary to ensure that potential solutions are not lost. Others argue that crossover in a largely uniform population only serves to propagate innovations originally found by mutation, and in a non-uniform population crossover is nearly always equivalent to a very large mutation (which is likely to be catastrophic).
- Though GAs can be used for global optimization in known intractable domains, GAs are not always good at finding optimal solutions. Their strength tends to be in rapidly locating *good* solutions, even for difficult search spaces.

Variants

The simplest algorithm represents each chromosome as a bit string. Typically, numeric parameters can be represented by integers, though it is possible to use floating point representations. The basic algorithm performs crossover and mutation at the bit level.

Other variants treat the chromosome as a list of numbers which are indexes into an instruction table, nodes in a linked list, hashes, objects, or any other imaginable data structure. Crossover and mutation are performed so as to respect data element boundaries. For most data types, specific variation operators can be designed. Different chromosomal data types seem to work better or worse for different specific problem domains.

Efficiency

Genetic algorithms are known to produce good results for some problems. Their major disadvantage is that they are relatively slow, being very computationally intensive compared to other methods, such as random optimization.

Recent speed improvements have focused on speciation, where crossover can only occur if individuals are closely-enough related.

Genetic algorithms are extremely easy to adapt to parallel computing and clustering environments. One method simply treats each node as a parallel population. Organisms are then migrated from one pool to another according to various propagation techniques.

Another method, the Farmer/worker architecture, designates one node the *farmer*, responsible for organism selection and fitness assignment, and the other nodes as *workers*, responsible for recombination, mutation, and function evaluation.

GA Approaches to Data Mining: Feature Selection, Variable Ordering, Clustering

A genetic program is ideal for implementing wrappers where parameters are naturally encoded as chromosomes such as bit strings or permutations. This is precisely the case with variable (feature subset) selection, where a bit string can denote membership in the subset. Both feature selection (also called *variable elimination*) and variable ordering are methods for inductive bias control where the input representation is changed from the default: the original set of variables in a predetermined or random order.

- **Wrappers for feature selection:** Yang and Honavar (1998) describe how to develop a basic GA for feature selection. With a genetic wrapper, we seek to evolve parameter values using the performance criterion of the overall learning system as fitness. The term *wrapper* refers to the inclusion of an inductive learning module, or inducer, that produces a trained model such as a decision tree; the GA "wraps" around this module by taking the output of each inducer applied to a projection of the data that is specified by an individual chromosome, and measuring the fitness of this output. That is, it validates the model on holdout or cross-validation data. The performance of the overall system depends on the quality of the validation data, the fitness criterion, and also the "wrapped" inducer. For example, Minaei-Bidgoli and Punch (2003) report good results for a classification problem in the domain of education (predicting student performance) using a GA in tandem with principal components analysis (PCA) for front-end feature extraction and *k*-nearest neighbor as the wrapped inducer.
- **Wrappers for variable ordering:** To encode a variable ordering problem using a GA, a chromosome must represent a permutation α that describes the order in which variables are considered. In Bayesian network structure learning, for example, the permutation dictates an ordering of candidate nodes (variables) in a graphical model whose structure is unknown.
- **Clustering:** In some designs, a GA is used to perform feature extraction as well. This can be a single integrated GA for feature selection and extraction or one specially designed to perform clustering as Maulika and Bandyopadhyay (2000) developed.

Many authors of GA-based wrappers have independently derived criteria that resemble *minimum description length* (*MDL*) estimators – that is, they seek to minimize model size and the sample complexity of input as well as maximize generalization accuracy.

An additional benefit of genetic algorithm-based wrappers is that it can automatically calibrate "empirically determined" constants such as the coefficients *a*, *b*, and *c* introduced in the previous section. As we noted, this can be done using individual training data sets rather than assuming that a single optimum exists for a large set of machine learning problems. This is preferable to empirically calibrating parameters as if a single "best mixture" existed. Even if a very large and representative corpus of data sets were used for this purpose, there is no reason to believe that there is a single *a posteriori* optimum for genetic parameters such as weight allocation to inferential loss, model complexity, and sample complexity of data in the variable selection wrapper.

Finally, genetic wrappers can "tune themselves" – for example, the GA-based inductive learning system of De Jong and Spears (1991) learns propositional rules from data and adjusts constraint parameters that control how these rules can be generalized. Mitchell notes (1997) that this is a method for evolving the learning strategy itself. Many classifier systems also implement performance-tuning wrappers in this way. Finally, population size and other constants for controlling elitism, niching, sharing, and scaling can be controlled using automatically tuned, or "parameterless", GAs.

GA Approaches to Data Warehousing and Data Modeling

Cheng, Lee, and Wong (2002) apply a genetic algorithm to perform vertical partitioning of a relational database via clustering. The objective is to minimize the number of page accesses (block transfers and seeks) while grouping some attributes of the original relations into frequently-used vertical class fragments. By treating the table of transactions versus attributes as a map to be divided into contiguous clusters, (Cheng, et al., 2002). are able to reformulate the vertical partitioning problem as a Traveling Salesman Problem, a classical *NP*-complete optimization problem where genetic algorithms have successfully been applied (Goldberg, 1989). Zhang, et al. (2001) use a query cost-driven genetic algorithm to optimize single composite queries, multiple queries with reusable relational expressions, and precomputed materialized views for data warehousing. The latter is treated as a feature selection problem represented using a simple GA with single-point crossover. Osinski (2005) uses a similar approach to select subcubes and construct indices as a step in query optimization. The goal for all of the above researchers was a database that responds efficiently to data warehousing and online analytical processing operations.

GA-Based Inducers

De Jong and Spears (1991) pioneered the use of GAs in rule induction, but work on integrating the various criteria for rule quality, including various interestingness measures, continues. Ghosh and Nath (2004) have recently explored the intrinsically multi-objective fitness of rules, which trade novelty against generality and applicability (coverage) and accuracy. Shenoy *et al.* (2005) show how GAs provide a viable adaptive method for dealing with rule induction from dynamic data sets — those that reflect frequent changes in consumer habits and buying patterns.

Problem domains

Problems which appear to be particularly appropriate for solution by genetic algorithms include timetabling and scheduling problems, and many scheduling software packages are based on GAs. GAs have also been applied to engineering and are often applied as an approach to solve global optimization problems. Because of their ability to deal with hard problems, particularly by exploring combinatorially large and complex search spaces rapidly, GAs have been seen by researchers such as Flockhart and Radcliffe (1996) as a potential guiding mechanism for knowledge discovery in databases, that can tune parameters such as interestingness measures, and "steer" a data mining system away from wasteful patterns and rules.

History

John Holland was the pioneering founder of much of today's work in genetic algorithms, which has moved on from a purely theoretical subject (though based on computer modelling) to provide methods which can be used to actually solve some difficult problems today.

Pseudo-code Algorithm

Choose initial population
Evaluate each individual's fitness
Repeat
    Select best-ranking individuals to reproduce
    Mate pairs at random
    Apply crossover operator
    Apply mutation operator
    Evaluate each individual's fitness
Until terminating condition (see below)

Terminating conditions often include:

- Fixed number of generations reached
- Budgeting: Allocated computation time/money used up
- An individual is found that satisfies minimum criteria
- The highest ranking individual's fitness is reaching or has reached a plateau such that successive iterations are not producing better results anymore.
- Manual inspection: May require start-and-stop ability
- Combinations of the above

FUTURE TRENDS

Related techniques

Genetic programming is a related technique developed by John Koza, in which computer programs, rather than

function parameters, are optimised. Genetic programming often uses tree-based internal data structures to represent the computer programs for adaptation instead of the list, or array, structures typical of genetic algorithms. Genetic programming algorithms typically require running time that is orders of magnitude greater than that for genetic algorithms, but they may be suitable for problems that are intractable with genetic algorithms.

CONCLUSION

Genetic algorithms provide a parallelizable and flexible mechanism for search and optimization that can be applied to several crucial problems in data warehousing and mining. These include feature selection, partitioning, and extraction (especially by clustering). However, as Goldberg (2002) notes, the science of genetic algorithms is still emerging, with new analytical methods ranging from schema theory to Markov chain analysis providing different ways to design working GA-based systems. The complex dynamics of genetic algorithms makes some engineers and scientists reluctant to use them for applications where convergence must be precisely guaranteed or predictable; however, genetic algorithms remain a very important general-purpose methodology that sometimes provides solutions where deterministic and classical stochastic methods fail. In data mining and warehousing, as in other domains where machine learning and optimization problems abound, genetic algorithms are useful because their fitness-driven design formulation makes them easy to automate and experiment with.

REFERENCES

Brameier, M. & Banzhaf, W. (2001). Evolving Teams of Predictors with Linear Genetic Programming. *Genetic Programming and Evolvable Machines 2*(4), 381-407.

Burke, E. K., Gustafson, S. & Kendall, G. (2004). Diversity in genetic programming: an analysis of measures and correlation with fitness. *IEEE Transactions on Evolutionary Computation,* 8(1), 47-62.

Cheng, C.-H., Lee, W.-K., & Wong, K.-F. (2002). A Genetic Algorithm-Based Clustering Approach for Database Partitioning. *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews*, 32(3), 215-230.

De Jong, K. A. & Spears, W. M. (1991). Learning concept classification rules using genetic algorithms. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI 1991)* (pp. 651-657).

Flockhart, I. W. & Radcliffe, N. J. (1996). A Genetic Algorithm-Based Approach to Data Mining. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pp. 299 - 302, Portland, or, August 2-4, 1996.

Ghosh, A. & Nath, B. (2004). Multi-Objective Rule Mining using Genetic Algorithms. *Information Sciences, Special issue on Soft Computing Data Mining*, 163(1-3), 123-133.

Goldberg, D. E. (1989), *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley.

Goldberg, D. E. (2002). *The Design of Innovation: Lessons from and for Competent Genetic Algorithms*. Norwell, MA: Kluwer.

Harvey, I. (1992), Species adaptation genetic algorithms: A basis for a continuing *SAGA*. In F.J. Varela and P. Bourgine (*eds.*) *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life* (pp. 346-354).

Keijzer, M. & Babovic, V. (2002). Declarative and Preferential Bias in GP-based Scientific Discovery. *Genetic Programming and Evolvable Machines 3*(1), 41-79.

Kishore, J. K., Patnaik, L. M., Mani, V. & Agrawal, V.K. (2000). Application of genetic programming for multicategory pattern classification. *IEEE Transactions on Evolutionary Computation 4*(3), 242-258.

Koza, J. (1992), *Genetic Programming: On the Programing of Computers by Means of Natural Selection*. Cambridge, MA: MIT Press.

Krawiec, K. (2002). Genetic Programming-based Construction of Features for Machine Learning and Knowledge Discovery Tasks. *Genetic Programming and Evolvable Machines 3*(4), 329-343.

Maulika, U., & Bandyopadhyay, S. (2000). Genetic Algorithm-Based Clustering Technique. *Pattern Recognition*, *33*(9), 1455-1465.

Minaei-Bidgoli, B. & Punch, W. F. (2003). Using genetic algorithms for data mining optimization in an educational web-based system. In *Proceedings of the Fifth Genetic and Evolutionary Computation Conference (GECCO 2003)*, 206-217.

Mitchell, M. (1996), *An Introduction to Genetic Algorithms*, Cambridge, MA: MIT Press.

Mitchell, T. (1997). Chapter 9: Genetic algorithms. *Machine Learning*. New York, NY: McGraw-Hill.

Muni, D. P., Pal, N. R. & Das, J. (2004). A novel approach to design classifiers using genetic programming. *IEEE Transactions on Evolutionary Computation 8*(2), 183-196.

Nikolaev, N. Y. & Iba, H. (2001). Regularization approach to inductive genetic programming. *IEEE Transactions on Evolutionary Computation 5*(4), p. 359-375.

Nikolaev, N. Y. & Iba, H. (2001). Accelerated genetic programming of polynomials. *Genetic Programming and Evolvable Machines 2*(3), 231-257.

Osinski, M. (2005). Optimizing a data warehouse using evolutionary computation. In *Proceedings of the Third International Atlantic Web Intelligence Conference (AWIC 2005)*, LNAI 3528. Lodz, Poland, June 2005.

Shenoy, P. D., Srinivasa, K. G., Venugopal, K. R., & Patnaik, L. M. (2005). Dynamic association rule mining using genetic algorithms. *Intelligent Data Analysis*, *9*(5), 439-453.

Yang, J. & Honavar, V. (1998). Feature subset selection using a genetic algorithm. *IEEE Intelligent Systems 13*(2), 44-49.

Zhang, C., Yao, X., & Yang, J. (2001). An evolutionary approach to materialized views selection in a data warehouse environment. In *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews*, *31*(3):282-294.

KEY TERMS

Chromosome: In genetic and evolutionary computation, a representation of the degrees of freedom of the system; analogous to a (stylized) biological chromosome, the threadlike structure which carries genes, the functional units of heredity.

Crossover: The sexual reproduction operator in genetic algorithms, typically producing one or two offspring from a selected pair of parents.

Evolutionary Computation: Solution approach guided by biological evolution, which begins with potential solution models, then iteratively applies algorithms to find the fittest models from the set to serve as inputs to the next iteration, ultimately leading to a model that best represents the data.

Mutation: One of two random search operators in genetic algorithms, or the sole one in selecto-mutative evolutionary algorithms, changing chromosomes at a randomly selected point.

Rule Induction: Process of learning, from cases or instances, if-then rule relationships consisting of an antecedent (if-part, defining the preconditions or coverage of the rule) and a consequent (then-part, stating a classification, prediction, or other expression of a property that holds for cases defined in the antecedent).

Selection: The means by which the fittest individuals are chosen and propagate from generation to generation in a GA.

Evolutionary Data Mining for Genomics

Laetitia Jourdan
University of Lille, France

Clarisse Dhaenens
University of Lille, France

El-Ghazali Talbi
University of Lille, France

INTRODUCTION

Knowledge discovery from genomic data has become an important research area for biologists. Nowadays, a lot of data is available on the web, but the corresponding knowledge is not necessarly also available. For example, the first draft of the human genome, which contains 3×10^9 letters, has been achieved in June 2000, but up to now only a small part of the hidden knowledge has been discovered. The aim of bioinformatics is to bring together biology, computer science, mathematics, statistics and information theory to analyze biological data for interpretation and prediction. Hence many problems encountered while studying genomic data may be modeled as data mining tasks, such as feature selection, classification, clustering, and association rule discovery.

An important characteristic of genomic applications is the large amount of data to analyze and it is, most of the time, not possible to enumerate all the possibilities. Therefore, we propose to model these knowledge discovery tasks as combinatorial optimization tasks, in order to apply efficient optimization algorithms to extract knowledge from large datasets. To design an efficient optimization algorithm, several aspects have to be considered. The main one is the choice of the type of resolution method according to the characteristics of the problem. Is it an easy problem, for which a polynomial algorithm may be found? If yes, let us design such an algorithm. Unfortunately, most of the time the response to the question is 'NO' and only heuristics, that may find good but not necessarily optimal solutions, can be used. In our approach we focus on evolutionary computation, which has already shown an interesting ability to solve highly complex combinatorial problems.

In this chapter, we will show the efficiency of such an approach while describing the main steps required to solve data mining problems from genomics with evolutionary algorithms. We will illustrate these steps with a real problem.

BACKGROUND

Evolutionary data mining for genomics groups three important fields: Evolutionary computation, knowledge discovery and genomics.

It is now well known that evolutionary algorithms are well suited for some data mining tasks and the reader may refer, for example, to (Freitas, 2008).

Here we want to show the interest of dealing with genomic data using evolutionary approaches. A first proof of this interest may be the book of Gary Fogel and David Corne on « Evolutionary Computation in Bioinformatics » which groups several applications of evolutionary computation to problems in the biological sciences, and in particular in bioinformatics (Corne, Pan, Fogel, 2008). In this book, several data mining tasks are addressed, such as feature selection or clustering, and solved thanks to evolutionary approaches.

Another proof of the interest of such approaches is the number of sessions around "Evolutionary computation in bioinformatics" in congresses on Evolutionary Computation. Man can take as an example, EvoBio, European Workshop on Evolutionary Computation and Machine Learning in Bioinformatics, or the special sessions on "Evolutionary computation in bioinformatics and computational biology" that have been organized during the last Congresses on Evolutionary Computation (CEC'06, CEC'07).

The aim of genomic studies is to understand the function of genes, to determine which genes are involved

in a given process and how genes are related. Hence experiments are conducted, for example, to localize coding regions in DNA sequences and/or to evaluate the expression level of genes in certain conditions. Resulting from this, data available for the bioinformatics researcher may not only deal with DNA sequence information but also with other types of data like for example in multi-factorial diseases the Body Mass Index, the sex, and the age. The example used to illustrate this chapter may be classified in this category.

Another type of data deals with the recent technology, called microarray, which allows the simultaneous measurement of the expression level of thousand of genes under different conditions (various time points of a process, absorption of different drugs…). This type of data requires specific data mining tasks as the number of genes to study is very large and the number of conditions may be limited. This kind of technology has now been extended to protein microarrays and generates also large amount of data. Classical questions are the classification or the clustering of genes based on their expression pattern, and commonly used approaches may vary from statistical approaches (Yeung & Ruzzo, 2001) to evolutionary approaches (Merz, 2002) and may use additional biological information such as the Gene Ontology - GO - (Speer, Spieth & Zell, 2004). A bi-clustering approach that allows the grouping of instances having similar characteristic for a subset of attributes (here, genes having the same expression patterns for a subset of conditions), has been applied to deal with this type of data and evolutionary approaches proposed (Bleuler, Prelié & Ziztler, 2004). Some authors are also working on the proposition of highly specialized crossover and mutation operators (Hernandez, Duval & Hao, 2007). In this context of microarray data analysis, data mining approaches have been proposed. Hence, for example, association rule discovery has been realized using evolutionary algorithms (Khabzaoui, Dhaenens & Talbi, 2004), as well as feature selection and classification (Jirapech-Umpai & Aitken, 2005).

MAIN THRUST OF THE CHAPTER

In order to extract knowledge from genomic data using evolutionary algorithms, several steps have to be considered:

1. Identification of the knowledge discovery task from the biological problem under study,
2. Design of this task as an optimization problem,
3. Resolution using an evolutionary approach.

In this part, we will focus on each of these steps. First we will present the genomic application we will use to illustrate the rest of the chapter and indicate the knowledge discovery tasks that have been extracted. Then we will show the challenges and some proposed solutions for the two other steps.

Genomics Application

The genomic problem under study is to formulate hypothesis on predisposition factors of different multifactorial diseases such as diabetes and obesity. In such diseases, one of the difficulties is that healthy people can become affected during their life so only the affected status is relevant. This work has been done in collaboration with the Biology Institute of Lille (IBL).

One approach aims to discover the contribution of environmental factors and genetic factors in the pathogenesis of the disease under study by discovering complex interactions such as [(gene A and gene B) or (gene C and environmental factor D)] in one or more population. The rest of the paper will take this problem as an illustration.

To deal with such a genomic application, the first thing is to formulate the underlying problem into a classical data mining task. This work must be done through discussions and cooperations with biologists in order to agree on the aim of the study. For example, in data of the problem under study, identifying groups of people can be modeled as a clustering task as we can not take into account non-affected people. Moreover a lot of attributes (features) have to be studied (a choice of 3652 points of comparison on the 23 chromosomes and two environmental factors) and classical clustering algorithms are not able to cope with so many features. So we decide to firstly execute a feature selection in order to reduce the number of loci into consideration and to extract the most influential features which will be used for the clustering. Hence, the model of this problem is decomposed into two phases: feature selection and clustering. The clustering is used to group individuals of same characteristics.

From a Data Mining Task to an Optimization Problem

One of the main difficulty in turning a data mining task into an optimization problem is to define the criterion to optimize. The choice of the optimization criterion which measures the quality of candidate knowledge to be extracted is very important and the quality of the results of the approach depends on it. Indeed, developing a very efficient method that does not use the right criterion will lead to obtain "the right answer to the wrong question"!! The optimization criterion can be either specific to the data mining task or dependent on the biological application, and different choices exist. For example, considering the gene clustering, the optimization criterion can be the minimization of the minimum sum-of-squares (MSS) (Merz, 2002) while for the determination of the members of a predictive gene group, the criterion can be the maximization of the classification success using a maximum likelihood (MLHD) classification method (Ooi & Tan, 2003).

Once the optimization criterion is defined, the second step of the design of the data mining task into an optimization problem is to define the encoding of a solution which may be independent of the resolution method. For example, for clustering problems in gene expression mining with evolutionary algorithm, Faulkenauer and Marchand (2001) used the specific CGA encoding that is dedicated to grouping problems and is well suited to clustering.

Regarding the genomic application used to illustrate the chapter, two phases are isolated. For the feature selection, an optimization approach is adopted, using an evolutionary algorithm (see next paragraph) whereas a classical approach (k-means) is chosen for the clustering phase (MacQueen, 1967). Determining the optimization criterion for the feature selection was not an easy task as it is difficult not to favor small sets of features. A corrective factor is introduced (Jourdan et al., 2002).

Solving with Evolutionary Algorithms

Once the formalization of the data mining task into an optimization problem is done, resolution methods can be either, exact methods, specific heuristics or metaheuristics. As the space of the potential knowledge is exponential in genomics problems (Zaki & Ho, 2000), exact methods are almost always discarded. The drawbacks of heuristic approaches are that it is difficult to cope with multiple solutions and not easy to integrate specific knowledge in a general approach. The advantages of metaheuristics are that you can define a general framework to solve the problem while specializing some agents in order to suit to a specific problem. Genetic Algorithms (GAs), which represent a class of evolutionary methods, have given good results on hard combinatorial problems (Michalewicz, 1996).

In order to develop a genetic algorithm for knowledge discovery, we have to focus on:

- Operators,
- Diversification mechanisms,
- Intensification mechanisms.

Operators allow GAs to explore the search space and must be adapted to the problem. There are commonly two classes of operators: mutation and crossover. The mutation allows diversity. For the feature selection task under study, the mutation flips *n* bits (Jourdan, Dhaenens & Talbi, 2001). The crossover produces one, two or more children solutions by recombining two or more parents. The objective of this mechanism is to keep useful information of the parents in order to ameliorate the solutions. In the considered problem, the subset-oriented common feature crossover operator (SSOCF) has been used. Its objective is to produce offsprings that have a distribution similar to their parents. This operator is well adapted for feature selection (Emmanouilidis, Hunter & MacIntyre, 2000). Another advantage of evolutionary algorithms is that you can easily use other data mining algorithms as an operator; for example a kmeans iteration may be used as an operator in a clustering problem (Krishma & Murty, 1999).

Working on knowledge discovery on a particular domain with optimization methods leads to the definition and the use of several operators, where some may use domain knowledge and some other may be specific to the model and/or the encoding. To take advantage of all the operators, the idea is to use adaptive mechanisms (Hong, Wang & Chen, 2000) that help to adapt application probabilities of these operators according to the progress they produce. Hence, operators that are less efficient are less used and this may change during the search if they become more efficient.

Diversification mechanisms are designed to avoid premature convergence. There exist several mecha-

nisms. The more classical are the sharing and the random immigrant. The sharing boosts the selection of individuals that lie in less crowded areas of the search space. To apply such a mechanism, a distance between solutions has to be defined. In the feature selection for the genomic association discovery, a distance has been defined by integrating knowledge of the application domain. The distance is correlated to a Hamming distance which integrates biological notions (chromosomal cut, inheritance notion…).

A further approach to the diversification of the population, is 'the random immigrant' that introduces new individuals. An idea is to generate new individuals by recording statistics on previous selections.

Assessing the efficiency of such algorithms applied to genomic data is not easy as most of the time, biologists have no exact idea about what must be found. Hence, one step in this analysis is to develop simulated data for which optimal results are known. In this manner it is possible to measure the efficiency of the proposed method. For example, for the problem under study, predetermined genomic associations were constructed to form simulated data. Then the algorithm has been tested on these data and was able to find these associations.

FUTURE TRENDS

There is much works on evolutionary data mining for genomics. In order to be more and more efficient and to propose more interesting solutions to biologists, the researchers are investigating multicriteria design of the data mining tasks. Indeed, we exposed that one of the critical phases was the determination of the optimization criterion, and it may be difficult to select a single one. As a response to this problem, the multicriteria design allows us to take into account some criteria dedicated to a specific data mining task and some criteria coming from the application domain. Evolutionary algorithms that work on a population of solutions, are well adapted to multicriteria problems as they can exploit Pareto approaches and propose several good solutions (solutions of best compromise).

For data mining in genomics, association rule discovery has not been very well applied, and should be studied carefully as this is a very general model. Moreover, an interesting multicriteria model has been proposed for this task and starts to give some interesting results by using multicriteria genetic algorithms (Jourdan et al., 2004).

CONCLUSION

Genomic is a real challenge for researchers in data mining as most of the problems encountered are difficult to address for several reasons: 1/ the objectives are not always very clear, neither for the data mining specialist, nor for the biologist and a real dialog has to exist between the two; 2/ data may be uncertain and noisy; 3/ the number of experiments may be not enough in comparison with the number of elements that have to be studied.

In this chapter we present the different phases required to deal with data mining problems arriving in genomics thanks to optimization approaches. The objective was to give, for each phase, the main challenges and to illustrate through an example some responses. We saw that the definition of the optimization criterion is a central point of this approach and the multicriteria optimization may be used as a response to this difficulty.

REFERENCES

Bleuler, S., Prelié, A., & Ziztler, E. (2004). An EA Framework for Biclustering of Gene expression data. *Congress on Evolutionary Computation (CEC04)*, Portland, USA, pp. 166-173.

Corne, D, Pan, Y., & Fogel, G. (2008). *Computational Intelligence in Bioinformatics*, IEEE Computer Society Press.

Emmanouilidis, C., Hunter, A., & MacIntyre, J. (2000). A Multiobjective Evolutionary Setting for Feature Selection and a Commonality-Based Crossover Operator. *Congress on Evolutionary Computation. (CEC00)*, San Diego, California, USA, vol. 1(2), pp. 309-316, IEEE, Piscataway, NJ, USA.

Falkenauer E. and Marchand A. (2001). Using k-Means? Consider ArrayMiner, In *Proceedings of the 2001 International Conference on Mathematics and Engineering Techniques in Medicine and Biological Sciences (METMBS'2001)*, Las Vegas, Nevada, USA.

Freitas, A.A. (2008). Soft Computing for Knowledge Discovery and Data Mining, chapter *A Review of evolutionary Algorithms for Data Mining*, Springer.

Hernandez Hernandez, J.C., Duval, B., & Hao, J-K. (2007). A genetic embedded approach for gene selection and classification of microarray data. *Lecture Notes in Computer Science* 4447: 90-101, Springer.

Hong, T-P., Wang, H-S., & Chen, W-C. (2000). Simultaneously Applying Multiple Mutation Operators in Genetic Algorithms. *J. Heuristics* 6(4), 439-455.

Jirapech-Umpa, T., & Aitken, S (2005). Feature selection and classification for microarray data analysis: Evolutionary methods for identifying predictive genes. *BMC Bioinformatics*, 6, 148.

Jourdan, L., Dhaenens, C., & Talbi, E.G. (2001). An Optimization Approach to Mine Genetic Data. *METMBS'01 Biological data mining and knowledge discovery*, pp. 40-46.

Jourdan, L., Dhaenens, C., Talbi, E.-G., & Gallina, S. (2002) A Data Mining Approach to Discover Genetic Factors involved in Multifactorial Diseases. *Knowledge Based Systems (KBS) Journal*, Elsevier Science, 15(4), 235-242.

Jourdan, L., Khabzaoui, M., Dhaenens, C., & Talbi, E-G. (2004). Handbook of Bioinspired Algorithms and Applications, chapter *A hybrid metaheuristic for knowledge discovery in microarray experiments*. CRC Press.

Khabzaoui, M., Dhaenens, C., & Talbi, E-G. (2004). Association rules discovery for DNA microarray data. *Bioinformatics Workshop of SIAM International Conference on Data Mining*, Orlando, USA, pp. 63-71.

Krishna, K., & Murty, M. (1999) Genetic K-means algorithm. IEEE Transactions on Systems, Man and Cybernetics—Part B: *Cybernetics, 29*(3), 433-439.

MacQueen, J. B. (1967). Some Methods for classification and Analysis of Multivariate Observations, *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, University of California Press, pp. 281-297.

Michalewicz, Z. (1996). *Genetic Algorithms + Data Strcutures = Evolution Programs.* Springer-Verlag.

Merz, P. (2002). Clustering gene expression profiles with memetic algorithms, in *Proc. 7th international conference on Parallel problem Solving from Nature (PPSN VII)*, LNCS, pp. 811-820.

Ooi, C. H., & Tan, P. (2003). Genetic algorithms applied to multi-class prediction for the analysis of gene expression data. *Bioinformatics*, 19, 37-44.

Speer, N., Spieth, C., & Zell, A. (2004). A Memetic Co-clustering Algorithm for Gene Expression Profiles and Biological Annotation. *Congress on Evolutionary Computation (CEC04)*, Portland, USA, pp. 1631-1638.

Yeung, K.Y., & Ruzzo, W.L. (2001). Principal Component analysis for clustering gene expression data. *Bioinformatics*, 17, 763-774.

Zaki, M., & Ho, C.T. (2000). *Large-Scale Parallel Data Mining*, volume 1759 of State-of-the-Art Survey, LNAI.

KEY TERMS

Association Rule Discovery: Implication of the form X → Y where X and Y are sets of items, that indicates that if the condition X is verified, the prediction may be Y valid according to quality criteria (support, confidence, surprise, …).

Bioinformatics: Field of science in which biology, computer science, and information technology merge into a single discipline.

Clustering: Data mining task in which the system has to group a set of objects without any information on the characteristics of the classes.

Crossover (in Genetic algorithm): Genetic operator used to vary the genotype of a chromosome or chromosomes from one generation to the next. It is analogous to reproduction and biological crossover, upon which genetic algorithms are based.

Feature (or Attribute): Quantity or quality describing an instance.

Feature Selection: Task of identifying and selecting a useful subset of features from a large set of redundant, perhaps irrelevant, features.

Genetic Algorithm: Evolutionary algorithm using a population and based on the Darwinian principle: the "survive of the fittests".

Multifactorial Disease: Disease caused by several factors. Often, multifactorial diseases are due to two kinds of causality which interact: one is genetic (and often polygenetic) and the other is environmental.

Mutation (in Genetic algorithm): Genetic operator used to maintain genetic diversity from one generation of a population of chromosomes to the next. It is analogous to biological mutation and it modifies a small part of the genotype.

Optimization Criterion: Measure that evaluates the quality of a solution of an optimization problem.

Evolutionary Development of ANNs for Data Mining

Daniel Rivero
University of A Coruña, Spain

Juan R. Rabuñal
University of A Coruña, Spain

Julián Dorado
University of A Coruña, Spain

Alejandro Pazos
University of A Coruña, Spain

INTRODUCTION

Artificial Neural Networks (ANNs) are learning systems from the Artificial Intelligence (AI) world that have been used for solving complex problems related to different aspects as classification, clustering, or regression (Haykin, 1999), although they have been specially used in Data Mining. These systems are, due to their interesting characteristics, powerful techniques used by the researchers in different environments (Rabuñal, 2005).

Nevertheless, the use of ANNs implies certain problems, mainly related to their development processes. The development of ANNs can be divided into two parts: architecture development and training and validation. The architecture development determines not only the number of neurons of the ANN, but also the type of the connections among those neurons. The training will determine the connection weights for such architecture.

Traditionally, and given that the architecture of the network depends on the problem to be solved, the architecture design process is usually performed by the use of a manual process, meaning that the expert has to test different architectures to find the one able to achieve the best results. Therefore, the expert must perform various tests for training different architectures in order to determine which one of these architectures is the best one. This is a slow process due to the fact that architecture determination is a manual process, although techniques for relatively automatic creation of ANNs have been recently developed.

This work presents various techniques for the development of ANNs, so that there would be needed much less human participation for such development.

BACKGROUND

The development of ANNs has been widely treated with very different techniques in AI. The world of evolutionary algorithms is not an exception, and proof of it is the large amount of works that have been published is this aspect using several techniques (Nolfi, 2002; Cantú-Paz, 2005). These techniques follow the general strategy of an evolutionary algorithm: an initial population with different types of genotypes encoding also different parameters – commonly, the connection weights and/or the architecture of the network and/or the learning rules – is randomly created. Such population is evaluated for determining the fitness of every individual. Subsequently, the population is repeatedly induced to evolve by means of different genetic operators (replication, crossover, mutation) until a certain termination parameter has been fulfilled (for instance, the achievement of an individual good enough or the accomplishment of a predetermined number of generations).

As a general rule, the field of ANNs generation using evolutionary algorithms is divided into three

main groups: evolution of weights, architectures and learning rules.

The evolution of weight starts from an ANN with an already determined topology. In this case, the problem to be solved is the training of the connection weights, attempting to minimise the network failure. Most of training algorithms, as backpropagation (BP) algorithm (Rumelhart, 1986), are based on gradient minimisation, which presents several inconveniences (Sutton, 1986). The main of these disadvantages is that, quite frequently, the algorithm gets stuck into a local minimum of the fitness function and it is unable to reach a global minimum. One of the options for overcoming this situation is the use of an evolutionary algorithm, so the training process is done by means of the evolution of the connection weights within the environment defined by both, the network architecture, and the task to be solved. In such cases, the weights can be represented either as the concatenation of binary values or of real numbers on a genetic algorithm (GA) (Greenwood, 1997). The main disadvantage of this type of encoding is the permutation problem. This problem means that the order in which weights are taken at the vector might cause that equivalent networks might correspond to completely different chromosomes, making the crossover operator inefficient.

The evolution of architectures includes the generation of the topological structure. This means establishing the connectivity and the transfer function of each neuron. The network architecture is highly important for the successful application of the ANN, since the architecture has a very significant impact on the processing ability of the network. Therefore, the network design, traditionally performed by a human expert using trial and error techniques on a group of different architectures, is crucial. In order to develop ANN architectures by means of an evolutionary algorithm it is needed to choose how to encode the genotype of a given network for it to be used by the genetic operators.

At the first option, direct encoding, there is a one-to-one correspondence between every one of the genes and their subsequent phenotypes (Miller, 1989). The most typical encoding method consists of a matrix that represents an architecture where every element reveals the presence or absence of connection between two nodes (Alba, 1993). These types of encoding are generally quite simple and easy to implement. However, they also have a large amount of inconveniences as scalability (Kitano, 1990), the incapability of encoding repeated structures, or permutation (Yao, 1998).

In comparison with direct encoding, there are some indirect encoding methods. In these methods, only some characteristics of the architecture are encoded in the chromosome and. These methods have several types of representation.

Firstly, the parametric representations represent the network as a group of parameters such as number of hidden layers, number of nodes for each layer, number of connections between two layers, etc (Harp, 1989). Although the parametric representation can reduce the length of the chromosome, the evolutionary algorithm performs the search within a restricted area in the search space representing all the possible architectures. Another non direct representation type is based on a representation system that uses the grammatical rules (Kitano, 1990). In this system, the network is represented by a group of rules, shaped as production rules that make a matrix that represents the network, which has several restrictions.

The growing methods represent another type of encoding. In this case, the genotype does not encode a network directly, but it contains a group of instructions for building up the phenotype. The genotype decoding will consist on the execution of those instructions (Nolfi, 2002), which can include neuronal migrations, neuronal duplication or transformation, and neuronal differentiation.

Another important non-direct codification is based on the use of fractal subsets of a map. According to Merrill (1991), the fractal representation of the architectures is biologically more plausible than a representation with the shape of rules. Three parameters are used which take real values to specify each node in an architecture: a border code, an entry coefficient and an exit code.

One important characteristic to bear in mind is that all those methods evolve architectures, either alone (most commonly) or together with the weights. The transfer function for every node of the architecture is supposed to have been previously fixed by a human expert and is the same for all the nodes of the network –or at least, all the nodes of the same layer–, despite of the fact that such function has proved to have a significant impact on network performance (DasGupta, 1992). Only few methods that also induce the evolution of the transfer function have been developed (Hwang, 1997).

With regards to the evolution of the learning rule, there are several approaches (Turney, 1996), although

most of them are only based on how learning can modify or guide the evolution and also on the relationship among the architecture and the connection weights. There are only few works focused on the evolution of the learning rule itself.

ANN DEVELOPMENT WITH GENETIC PROGRAMMING

This section very briefly shows an example of how to develop ANNs using an AI tool, Genetic Programming (GP), which performs an evolutionary algorithm, and how it can be applied to Data Mining tasks.

Genetic Programming

GP (Koza, 92) is based on the evolution of a given population. Its working is similar to a GA. In this population, every individual represents a solution for a problem that is intended to be solved. The evolution is achieved by means of the selection of the best individuals – although the worst ones have also a little chance of being selected – and their mutual combination for creating new solutions. This process is developed using selection, crossover and mutation operators. After several generations, the population is expected to contain some good solutions for the problem.

The GP encoding for the solutions is tree-shaped, so the user must specify which are the terminals (leaves of the tree) and the functions (nodes capable of having descendants) for being used by the evolutionary algorithm in order to build complex expressions. These can be mathematical (including, for instance, arithmetical or trigonometric operators), logical (with Boolean or relational operators, among others) or other type of even more complex expressions.

The wide application of GP to various environments and its consequent success are due to its capability for being adapted to numerous different problems. Although the main and more direct application is the generation of mathematical expressions (Rivero, 2005), GP has been also used in others fields such as filter design (Rabuñal, 2003), knowledge extraction (Rabuñal, 2004), image processing (Rivero, 2004), etc.

Model Overview

The GP-development of ANNs is performed by means of the GP typing property (Montana, 1995). This property provides the ability of developing structures that follow a specific grammar with certain restrictions. In order to be able to use GP to develop any kind of system, it is necessary to specify the set of operators that will be in the tree. With them, the evolutionary system must be able to build correct trees that represent ANNs. An overview of the operators used here can be seen on Table 1.

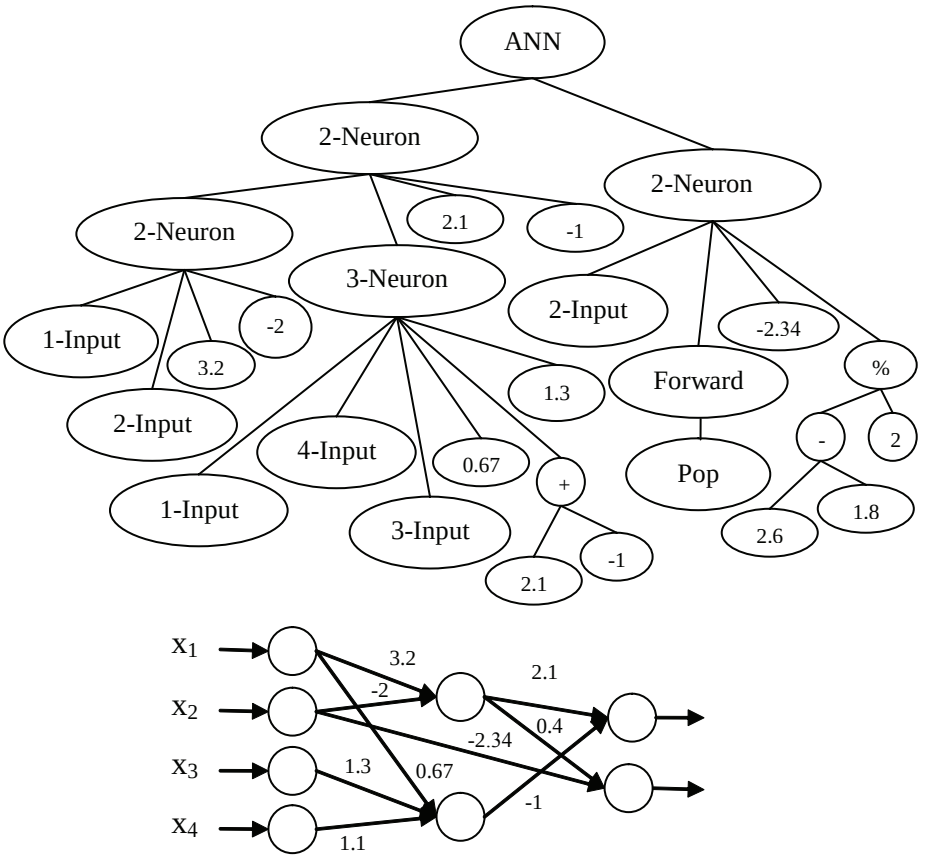
This table shows a summary of the operators that can be used in the tree. This set of terminals and functions are used to build a tree that represents an ANN. Although these sets are not explained in the text, in Fig. 1 can be seen an example of how they can be used to represent an ANN.

These operators are used to build GP trees. These trees have to be evaluated, and, once the tree has been evaluated, the genotype turns into phenotype. In other words, it is converted into an ANN with its weights

Table 1. Summary of the operators to be used in the tree

Node	Type	Num. Children	Children type
ANN	ANN	Num. outputs	NEURON
n-Neuron	NEURON	2*n	n NEURON n REAL (weights)
n-Input	NEURON	0	-
+,-,*,%	REAL	2	REAL
[-4.4]	REAL	0	-
Forward	NEURON	1	NEURON
Pop	NEURON	0	-

Figure. 1. GP tree and its resulting network



already set (thus it does not need to be trained) and therefore can be evaluated. The evolutionary process demands the assignation of a fitness value to every genotype. Such value is the result of the evaluation of the network with the pattern set that represents the problem. This result is the Mean Square Error (MSE) of the difference between the network outputs and the desired outputs. Nevertheless, this value has been modified in order to induce the system to generate simple networks. The modification has been made by adding a penalization value multiplied by the number of neurons of the network. In such way, and given that the evolutionary system has been designed in order to minimise an error value, when adding a fitness value, a larger network would have a worse fitness value. Therefore, the existence of simple networks would be preferred as the penalization value that is added is proportional to the number of neurons at the ANN. The calculus of the final fitness will be as follows:

$$fitness = MSE + N * P$$

Where N is the number of neurons of the network and P is the penalization value for such number.

Example of Applications

This technique has been used for solving problems of different complexity taken from the UCI (Mertz, 2002). All these problems are knowledge-extraction problems from databases where, taking certain features as a basis, it is intended to perform a prediction about another attribute of the database. A small summary of the problems to be solved can be seen at Table 2.

All these databases have been normalised between 0 and 1 and divided into two parts, taking the 70% of the data base for training and using the remaining 30% for performing tests.

Preliminary Results

Several experiments have been performed in order to evaluate the system performance with regards to data mining. The values taken for the parameters at these experiments were the following:

- Crossover rate: 95%.
- Mutation probabillity: 4%.
- Selection algorithm: 2-individual tournament.
- Creation algorithm: Ramped Half&Half.
- Tree maximum height: 6.
- Maximum inputs for each neuron: 12.

Table 3 shows a comparison of the results obtained for different penalization values. The values range from very high (0.1) to very small (0.00001,0). High values only enables the creation of very small networks with a subsequent high error which enables the creation of large networks, and low values lead to overfitting problem. This overfitting can be noticed at the table in the training error decrease together with a test error increase.

The number of neurons as well as of connections that were obtained at the resulting networks is also shown in table 3. Logically, such number is higher as the penalization increases. The results correspond to the MSE obtained after both, the training and the test of every problem. As it can be observed, the results clearly prove that the problems have been satisfactorily solved and, as far as penalization parameter is concerned, intermediate values are preferred for the creation of

Table 2. Summary of the problems to be solved

	Number of inputs	Number of instances	Number of outputs
Breast Cancer	9	699	1
Iris Flower	4	150	3
Ionosphere	34	351	1

Table 3. Results with different penalization values

		Penalization					
		0.1	0.01	0.001	0.0001	0.00001	0
Breast Cancer	Neurons	1.75	2	2.65	9.3	22.65	27.83
	Connections	8.8	9.2	13.1	47.6	100.7	126.94
	Training	0.03371	0.01968	0.01801	0.01426	0.01366	0.01257
	test	0.03392	**0.02063**	0.02096	0.02381	0.02551	0.02514
Iris Flower	Neurons	3	4	8.45	24.2	38.9	42.85
	Connections	8.8	11.95	27.85	86.55	140.25	157.05
	Training	0.06079	0.03021	0.01746	0.01658	0.01681	0.01572
	test	0.07724	0.05222	**0.03799**	0.04084	0.04071	0.04075
Ionosphere	Neurons	1	2.35	8.15	21.6	32.4	42.45
	Connections	6.1	11.95	48.3	128.15	197.4	261.8
	Training	0.09836	0.06253	0.03097	0.02114	0.02264	0.01781
	test	0.10941	0.09127	**0.06854**	0.07269	0.07393	0.07123

networks. These intermediate values in the parameter will allow the creation of networks large enough for solving the problem, avoiding over-fitting, although it should be changed for every problem.

FUTURE TRENDS

The future line of works in this area would be the study of the rest of system parameters in order to evaluate their impact on the results from different problems.

Another interesting line consists on the modification of the GP algorithm for its operation with graphs instead of trees. In such way, the neurons connectivity will be reflected more directly in the genotype, with no need of special operators, therefore positively affecting to the system efficiency.

CONCLUSION

This work presents an overview of techniques in which EC tools are used to develop ANNs. It also presents a new method for ANN creation by means of PG. It is described how the GP algorithm can be configured for the development of simpler networks.

The results presented here are preliminary, since it is needed the performance of more experiments in order to study the effect of the different parameters attending to the complexity of the problem to be solved. In any case, despite using the same parameter values for all the problems during the experiments described here, the results have been quite good.

REFERENCES

Alba E., Aldana J.F., & Troya J.M. (1993). *Fully automatic ANN design: A genetic approach*. Proc. Int. Workshop Artificial Neural Networks (IWANN'93), Lecture Notes in Computer Science. 686. Berlin, Germany: Springer-Verlag, 686, 399-404.

Cantú-Paz E.. & Kamath C. (2005). An Empirical Comparison of Combinatios of Evolutionary Algorithms and Neural Networks for Classification Problems. I*EEE Transactions on systems, Man and Cybernetics – Part B: Cybernetics*, 915-927.

DasGupta, B. & Schnitger, G. (1992). *Efficient approximation with neural networks: A comparison of gate functions*. Dep. Comput. Sci., Pennsylvania State Univ., University Park, Tech. Rep.

Greenwood, G.W. (1997). Training partially recurrent neural networks using evolutionary strategies. *IEEE Trans. Speech Audio Processing*, *5*, 192-194.

Harp, S.A., Samad, T., & Guha, A. (1989). *Toward the genetic synthesis of neural networks*. Proc. 3rd Int. Conf. Genetic Algorithms and Their Applications, J.D. Schafer, Ed. San Mateo, CA: Morgan Kaufmann. 360-369.

Haykin, S. (1999). *Neural Networks* (2nd ed.). Englewood Cliffs, NJ: Prentice Hall.

Hwang, M.W., Choi J.Y., & Park J. (1997). Evolutionary projection neural networks. *Proc. 1997 IEEE Int. Conf. Evolutionary Computation*, ICEC'97. 667-671.

Jung-Hwan, K., Sung-Soon, C., & Byung-Ro, M. (2005). Normalization for neural network in genetic search. *Genetic and Evolutionary Computation Conference*, 1-10.

Kitano, H. (1990) Designing neural networks using genetic algorithms with graph generation system. Complex Systems, 4, 461-476.

Koza, J. R. (1992) Genetic Programming: On the Programming of Computers by Means of Natural Selection. Cambridge, MA: MIT Press.

Mertz, C.J., & Murphy, P.M. (2002). *UCI repository of machine learning databases*. http://www-old.ics.uci.edu/pub/machine-learning-databases.

Merrill, J.W.L., & Port, R.F. (1991). Fractally configured neural networks. *Neural Networks, 4*(1), 53-60.

Miller, G.F., Todd, P.M., & Hedge, S.U. (1989). *Designing neural networks using genetic algorithms*. Proceedings of the Third International Conference on Genetic algorithms. San Mateo, CA: Morgan Kaufmann, 379-384.

Montana, D.J. (1995). Strongly typed genetic programming. *Evolutionary Computation, 3*(2), 199-200.

Nolfi S. & Parisi D. (2002) Evolution of Artificial Neural Networks. *Handbook of brain theory and neural networks*, Second Edition. Cambridge, MA: MIT Press. 418-421.

Rabuñal, J.R., Dorado, J., Puertas, J., Pazos, A., Santos, A., & Rivero, D. (2003) Prediction and Modelling of the Rainfall-Runoff Transformation of a Typical Urban Basin using ANN and GP. *Applied Artificial Intelligence.*

Rabuñal, J.R., Dorado, J., Pazos, A., Pereira, J., & Rivero, D. (2004). A New Approach to the Extraction of ANN Rules and to Their Generalization Capacity Through GP. *Neural Computation, 16*(7), 1483-1523.

Rabuñal, J.R. & Dorado J. (2005). *Artificial Neural Networks in Real-Life Applications*. Idea Group Inc.

Rivero, D., Rabuñal, J.R., Dorado, J. & Pazos, A. (2004). *Using Genetic Programming for Character Discrimination in Damaged Documents.* Applications of Evolutionary Computing, EvoWorkshops2004: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoMUSART, EvoSTOC (Conference proceedings). 349-358.

Rivero D., Rabuñal J.R., Dorado J., & Pazos A. (2005). *Time Series Forecast with Anticipation using Genetic Programming*. IWANN 2005. 968-975.

Rumelhart D.E., Hinton G.E., & Williams R.J. (1986). Learning internal representations by error propagation. Parallel Distributed Processing: Explorations in the Microstructures of Cognition. D. E. Rumelhart & J.L. McClelland, Eds. Cambridge, MA: MIT Press. 1, 318-362.

Sutton, R.S. (1986). Two problems with backpropagation and other steepest-descent learning procedure for networks. P*roc. 8th Annual Conf. Cognitive Science Society*. Hillsdale, NJ: Erlbaum. 823-831.

Turney, P., Whitley, D. & Anderson, R. (1996). Special issue on the baldwinian effect. *Evolutionary Computation. 4*(3), 213-329.

Yao, X. & Liu, Y. (1998). Toward designing artificial neural networks by evolution. *Appl. Math. Computation. 91*(1), 83-90.

Yao, X. (1999) Evolving artificial neural networks. *Proceedings of the IEEE, 87*(9), 1423-1447.

KEY TERMS

Area of the Search Space: Set of specific ranges or values of the input variables that constitute a subset of the search space.

Artificial Neural Networks: A network of many simple processors ("units" or "neurons") that imitates a biological neural network. The units are connected by unidirectional communication channels, which carry numeric data. Neural networks can be trained to find nonlinear relationships in data, and are used in applications such as robotics, speech recognition, signal processing or medical diagnosis.

Back-Propagation Algorithm: Learning algorithm of ANNs, based on minimising the error obtained from the comparison between the outputs that the network gives after the application of a set of network inputs and the outputs it should give (the desired outputs).

Data Mining: The application of analytical methods and tools to data for the purpose of identifying patterns, relationships or obtaining systems that perform useful tasks such as classification, prediction, estimation, or affinity grouping.

Evolutionary Computation: Solution approach guided by biological evolution, which begins with potential solution models, then iteratively applies algorithms to find the fittest models from the set to serve as inputs to the next iteration, ultimately leading to a model that best represents the data.

Genetic Programming: Machine learning technique that uses an evolutionary algorithm in order to optimise the population of computer programs according to a fitness function which determines the capability of a program for performing a given task.

Search Space: Set of all possible situations of the problem that we want to solve could ever be in.

Genotype: The representation of an individual on an entire collection of genes which the crossover and mutation operators are applied to.

Phenotype: Expression of the properties coded by the individual's genotype.

Population: Pool of individuals exhibiting equal or similar genome structures, which allows the application of genetic operators.

Evolutionary Mining of Rule Ensembles

Jorge Muruzábal
University Rey Juan Carlos, Spain

INTRODUCTION

Ensemble rule based classification methods have been popular for a while in the machine-learning literature (Hand, 1997). Given the advent of low-cost, high-computing power, we are curious to see how far can we go by repeating some basic learning process, obtaining a variety of possible inferences, and finally basing the global classification decision on some sort of ensemble summary. Some general benefits to this idea have been observed indeed, and we are gaining wider and deeper insights on exactly why this is the case in many fronts of interest.

There are many ways to approach the ensemble-building task. Instead of locating ensemble members independently, as in Bagging (Breiman, 1996), or with little feedback from the joint behavior of the forming ensemble, as in Boosting (see, e.g., Schapire & Singer, 1998), members can be created at random and then made subject to an evolutionary process guided by some fitness measure. Evolutionary algorithms mimic the process of natural evolution and thus involve populations of individuals (rather than a single solution iteratively improved by hill climbing or otherwise). Hence, they are naturally linked to ensemble-learning methods. Based on the long-term processing of the data and the application of suitable evolutionary operators, fitness landscapes can be designed in intuitive ways to prime the ensemble's desired properties. Most notably, beyond the intrinsic fitness measures typically used in pure optimization processes, fitness can also be endogenous, that is, it can prime the context of each individual as well.

BACKGROUND

A number of evolutionary mining algorithms are available nowadays. These algorithms may differ in the nature of the evolutionary process or in the basic models considered for the data or in other ways. For example, approaches based on the genetic programming (GP), evolutionary programming (EP), and classifier system (CS) paradigms have been considered, while predictive rules, trees, graphs, and other structures been have evolved. See Eiben and Smith (2003) for an introduction to the general GP, EP, and CS frameworks and Koza, Keane, Streeter, Mydlowec, Yu, and Lanza (2003) for an idea of performance by GP algorithms at the patent level. Here I focus on ensemble –rule based methods for classification tasks or supervised learning (Hand, 1997).

The CS architecture is naturally suitable for this sort of rule assembly problems, for its basic representation unit is the rule, or classifier (Holland, Holyoak, Nisbett, & Thagard, 1986). Interestingly, tentative ensembles in CS algorithms are constantly tested for successful cooperation (leading to correct predictions). The fitness measure seeks to reinforce those classifiers leading to success in each case. However interesting, the CS approach in no way exhausts the scope of evolutionary computation ideas for ensemble-based learning; see, for example, Kuncheva and Jain (2000), Liu, Yao, and Higuchi (2000), and Folino, Pizzuti, and Spezzano (2003).

Ensembles of trees or rules are the natural reference for evolutionary mining approaches. Smaller trees, made by rules (leaves) with just a few tests, are of particular interest. Stumps place a single test and constitute an extreme case (which is nevertheless used often). These rules are more general and hence tend to make more mistakes, yet they are also easier to grasp and explain. A related notion is that of support, the estimated probability that new data satisfy a given rule's conditions. A great deal of effort has been done in the contemporary CS literature to discern the idea of adequate generality, a recurrent topic in the machine-learning arena.

MAIN THRUST

Evolutionary and Tree-Based Rule Ensembles

In this section, I review various methods for ensemble formation. As noted earlier, in this article, I use the ensemble to build averages of rules. Instead of averaging, one could also select the most suitable classifier in each case and make the decision on the basis of that rule alone (Hand, Adams, & Kelly, 2001). This alternative idea may provide additional insights of interest, but I do not analyze it further here.

It is conjectured that maximizing the degree of interaction amongst the rules already available is critical for efficient learning (Kuncheva & Jain, 2000; Hand et al., 2001). A fundamental issue concerns then the extent to which tentative rules work together and are capable of influencing the learning of new rules. Conventional methods like Bagging and Boosting show at most moderate amounts of interaction in this sense. While Bagging and Boosting are useful, well-known data-mining tools, it is appropriate to explore other ensemble-learning ideas as well. In this article, I focus mainly on the CS algorithm. CS approaches provide interesting architectures and introduce complex nonlinear processes to model prediction and reinforcement. I discuss a specific CS algorithm and show how it opens interesting pathways for emergent cooperative behaviour.

Conventional Rule Assembly

In Bagging methods, different training samples are created by bootstraping, and the same basic learning procedure is applied on each bootstrapped sample. In Bagging trees, predictions are decided by majority voting or by averaging the various opinions available in each case. This idea is known to reduce the basic instability of trees (Breiman, 1996).

A distinctive feature of the Boosting approach is the iterative calling to a basic weak learner (WL) algorithm (Schapire & Singer, 1998). Each time the WL is invoked, it takes as input the training set — together with a dynamic (probability) weight distribution over the data — and returns a single tree. The output of the algorithm is a weighted sum itself, where the weights are proportional to individual performance error. The WL learning algorithm needs only to produce moderately successful models. Thus, trees and simplified trees (stumps) constitute a popular choice. Several weight updating schemes have been proposed. Schapire and Singer update weights according to the success of the last model incorporated, whereas in their LogitBoost algorithm, Friedman, Hastie, and Tibshirani (2000) let the weights depend on overall probabilistic estimates. This latter idea better reflects the joint work of all classifiers available so far and hence should provide a more effective guide for the WL in general.

The notion of abstention brings a connection with the CS approach that will be apparent as I discuss the match set idea in the followins sections. In standard boosting trees, each tree contributes a leaf to the overall prediction for any new x input data vector, so the number of expressing rules is the number of boosting rounds independently of x. In the system proposed by Cohen and Singer (1999), the WL essentially produces rules or single leaves C (rather than whole trees). Their classifiers are then maps taking only two values, a real number for those x verifying the leaf and 0 elsewhere. The final boosting aggregation for x is thus unaffected by all abstaining rules (with $x \notin C$), so the number of expressing rules may be a small fraction of the total number of rules.

The General CS-Based Evolutionary Approach

The general classifier system (CS) architecture invented by John Holland constitutes perhaps one of the most sophisticated classes of evolutionary computation algorithms (Hollandet et al., 1986). Originally conceived as a model for cognitive tasks, it has been considered in many (simplified) forms to address a number of learning problems. The nowadays standard stimulus-response (or single-step) CS architecture provides a fascinating approach to the representation issue. Straightforward rules (classifiers) constitute the CS building blocks. CS algorithms maintain a population of such predictive rules whose conditions are hyperplanes involving the wild-card character #. If we generalize the idea of hyperplane to mean "conjunctions of conditions on predictors where each condition involves a single predictor," We see that these rules are also used by many other learning algorithms. Undoubtedly, hyperplane interpretability is a major factor behind this popularity.

Critical subsystems in CS algorithms are the performance, credit-apportionment, and rule discovery modules (Eiben & Smith, 2003). As regards credit-apportionment, the question has been recently raised about the suitability of endogenous reward schemes, where *endogenous* refers to the overall context in which classifiers act, versus other schemes based on intrinsic value measures (Booker, 2000). A well-known family of algorithms is XCS (and descendants), some of which have been previously advocated as data-mining tools (see, e.g., Wilson, 2001). The complexity of the CS dynamics has been analyzed in detail in Westerdale (2001).

The match set $M=M(x)$ is the subset of matched (concurrently activated) rules, that is, the collection of all classifiers whose condition is verified by the input data vector x. The (point) prediction for a new x will be based exclusively on the information contained in this ensemble M.

A System Based on Support and Predictive Scoring

Support is a familiar notion in various data-mining scenarios. There is a general trade-off between support and predictive accuracy: the larger the support, the lower the accuracy. The importance of explicitly bounding support (or deliberately seeking high support) has been recognized often in the literature (Greene & Smith, 1994; Friedman & Fisher, 1999; Muselli & Liberati, 2002). Because the world of generality intended by using only high-support rules introduces increased levels of uncertainty and error, statistical tools would seem indispensable for its proper modeling. To this end, classifiers in the BYPASS algorithm (Muruzábal, 2001) differ from other CS alternatives in that they enjoy (support-minded) probabilistic predictions (thus extending the more common single-label predictions). Support plays an outstanding role: A minimum support level b is input by the analyst at the outset, and consideration is restricted to rules with (estimated) support above b. The underlying predictive distributions, R, are easily constructed and coherently updated following a standard Bayesian Multinomial-Dirichlet process, whereas their toll on the system is minimal memory-wise. Actual predictions are built by first averaging the matched predictive distributions and then picking the maximum a posteriori label of the result. Hence, by promoting mixtures of probability distributions, the BYPASS algorithm connects readily with mainstream ensemble-learning methods.

We can find sometimes perfect regularities, that is, subsets C for which the conditional distribution of the response Y (given $X \in C$) equals 1 for some output class: $P(Y=j \mid X \in C)=1$ for some j and 0 elsewhere. In the well-known multiplexer environment, for example, there exists a set of such classifiers such that 100% performance can be achieved. But in real situations, it will be difficult to locate strictly neat C unless its support is quite small. Moreover, putting too much emphasis on error-free behavior may increase the risk of overfitting; that is, we may infer rules that do not apply (or generalize poorly) over a test sample. When restricting the search to high-support rules, probability distributions are well equipped to represent high-uncertainty patterns. When the largest $P(Y=j \mid X \in C)$ is small, it may be especially important to estimate $P(Y=j \mid X \in C)$ for all j.

Furthermore, the use of probabilistic predictions $R(j)$ for $j=1, ..., k$, where k is the number of output classes, makes possible a natural ranking of the $M=M(x)$ assigned probabilities $R_i(y)$ for the true class y related to the current x. Rules i with large $R_i(y)$ (scoring high) are generally preferred in each niche. In fact, only a few rules are rewarded at each step, so rules compete with each other for the limited amount of resources. Persistent lack of reward means extinction — to survive, classifiers must get reward from time to time. Note that newly discovered, more effective rules with even better scores may cut dramatically the reward given previously to other rules in certain niches. The fitness landscape is thus highly dynamic, and lower scores $p_i(y)$ may get reward and survive provided they are the best so far at some niche. An intrinsic measure of fitness for a classifier $C \rightarrow R$ (such as the lifetime average score -log $R(Y)$, where Y is conditioned by $X \in C$) could hardly play the same role.

It is worth noting that BYPASS integrates three learning modes in its classifiers: Bayesian at the data-processing level, reinforcement at the survival (competition) level, and genetic at the rule-discovery (exploration) level. Standard genetic algorithms (GAs) are triggered by system failure and act always circumscribed to M. Because the Bayesian updating guarantees that in the long run, predictive distributions R reflect the true conditional probabilities $P(Y=j \mid X \in C)$, scores

Table 1. The skeleton BYPASS algorithm

Input parameters (h: reward intensity) (u: minimum utility)
Initialize classifier population, say P
Iterate for a fixed number of cycles:
- Sample training item (x,y) from file
- Build match set M=M(x) including all rules with x ? C
- Build point prediction y*
- Check for success y* = y
  - If successful, reward the h ≥2 highest scores $R_i(y)$ in M
  - If unsuccessful, add a new rule to P via a GA or otherwise
- Update predictive distributions R (in M)
- Update utility counters (in P)
- Check for low utility via threshold u and possibly delete some rules

become highly reliable to form the basis of learning engines such as reward or crossover selection. The BYPASS algorithm is sketched in Table 1. Note that utility reflects accumulated reward (Muruzábal, 2001). Because only matched rules get reward, high support is a necessary (but not sufficient) condition to have high utility. Conversely, low-uncertainty regularities need to comply with the (induced) bound on support. The background generalization rate $P_{\#}$ (controlling the number of #s in the random *C* built along the run) is omitted for clarity, although some tuning of $P_{\#}$ with regard to threshold *u* is often required in practice.

BYPASS has been tested on various tasks under demanding *b* (and not very large *h*), and the results have been satisfactory in general. Comparative smaller populations are used, and low uncertainty but high support rules are uncovered. Very high values for $P_{\#}$ (as high as 0.975) have been tested successfully in some cases (Muruzábal, 2001). In the juxtaposed (or concatenated) multiplexer environment, BYPASS is shown to maintain a compact population of relatively high uncertainty rules that solves the problem by bringing about appropriate match sets (nearly) all the time. Recent work by Butz, Goldberg, and Tharakunnel (2003) shows that XCS also solves this problem, although working at a lower level of support (generality).

To summarize, BYPASS does not relay on rule plurality for knowledge encoding because it uses compact probabilistic predictions (bounded by support). It requires no intrinsic value for rules and no added tailor-made heuristics. Besides, it tends to keep population size under control (with increased processing speed and memory savings). The ensembles (match sets) derived from evolution in BYPASS have shown good promise of cooperation.

FUTURE TRENDS

Quick interactive data-mining algorithms and protocols are nice when human judgment is available. When not, computer-intensive, autonomous algorithms capable of thoroughly squeezing the data are also nice for preliminary exploration and other purposes. In a sense, we should tend to rely on the latter to mitigate the nearly ubiquitous data overflow problem. Representation schemes and learning engines are crucial to the success of these unmanned agents and need, of course, further investigation. Ensemble methods have lots of appealing features and will be subject to further analysis and testing. Evolutionary algorithms will continue to uprise and succeed in yet some other application areas. Additional commercial spin-offs will keep coming. Although great progress has been made in identifying many key insights in the CS framework, some central points still need further discussion. Specifically, the idea of rules that perform its prediction following some kind of more elaborate computation is appealing, and indeed more functional representations of classifiers (such as multilayer perceptrons) have been proposed in the CS literature (see, e.g., Bull, 2002). On the theoretical side, a formal framework for more rigorous analysis in high-support learning is much needed. The task is not easy, however, because the target is somewhat more vague, and individual as well as collective interests should be brought to terms when evaluating the generality and uncertainty associated to rules. Also, further research should be conducted to clearly delineate the strengths of the various CS approaches against current alternative methods for rule ensemble formation and data mining.

CONCLUSION

Evolutionary rule mining is a successful, promising research area. Evolutionary algorithms constitute by now a very useful and wide class of stochastic optimization methods. The evolutionary CS approach is likely to provide interesting insights and cross-fertilization of ideas with other data-mining methods. The BYPASS algorithm discussed in this article has been shown to tolerate the high support constraint well, leading to pleasant and unexpected results in some problems. These results stress the latent predictive power of the ensembles formed by high uncertainty rules.

REFERENCES

Booker, L. B. (2000). Do we really need to estimate rule utilities in classifier systems? *Lecture Notes in Artificial Intelligence, 1813,* 125-142.

Breiman, L. (1996). Bagging predictors. *Machine Learning, 24*, 123-140.

Bull, L. (2002). On using constructivism in neural classifier systems. *Lecture Notes in Computer Science, 2439,* 558-567.

Butz, M. V., Goldberg, D. E., & Tharakunnel, K. (2003). Analysis and improvement of fitness exploitation in XCS: Bounding models, tournament selection, and bilateral accuracy. *Evolutionary Computation, 11*(3), 239-277.

Cohen, W. W., & Singer, Y. (1999). A simple, fast, and effective rule learner. *Proceedings of the 16th National Conference on Artificial Intelligence,*.

Eiben, A. E., & Smith, J. E. (2003). *Introduction to evolutionary computing*. Springer.

Folino, G., Pizzuti, C., & Spezzano, G. (2003). Ensemble techniques for parallel genetic programming based classifiers. *Lecture Notes in Computer Science, 2610,* 59-69.

Friedman, J. H., & Fisher, N. (1999). Bump hunting in high-dimensional data. *Statistics and Computing, 9*(2), 1-20.

Friedman, J. H., Hastie, T., & Tibshirani, R. (2000). Additive logistic regression: A statistical view of boosting. *Annals of Statistics, 28*(2), 337-407.

Greene, D. P., & Smith, S. F. (1994). Using coverage as a model building constraint in learning classifier systems. *Evolutionary Computation, 2*(1), 67-91.

Hand, D. J. (1997). *Construction and assessment of classification rules*. Wiley.

Hand, D. J., Adams, N. M., & Kelly, M. G. (2001). Multiple classifier systems based on interpretable linear classifiers. *Lecture Notes in Computer Science, 2096,* 136-147.

Holland, J. H., Holyoak, K. J., Nisbett, R. E., & Thagard, P. R. (1986). *Induction: Processes of inference, learning and discovery.* MIT Press.

Koza, J. R., Keane, M. A., Streeter, M. J., Mydlowec, W., Yu, J., & Lanza, G. (Eds.). (2003). *Genetic programming IV: Routine human-competitive machine intelligence*. Kluwer.

Kuncheva, L. I., & Jain, L. C. (2000). Designing classifier fusion systems by genetic algorithms. *IEEE Transactions on Evolutionary Computation, 4*(4), 327-336.

Liu, Y., Yao, X., & Higuchi, T. (2000). Evolutionary ensembles with negative correlation learning. *IEEE Transactions on Evolutionary Computation, 4*(4), 380-387.

Muruzábal, J. (2001). Combining statistical and reinforcement learning in rule-based classification. *Computational Statistics, 16*(3), 341-359.

Muselli, M., & Liberati, D. (2002). Binary rule generation via hamming clustering. *IEEE Transactions on Knowledge and Data Engineering, 14*, 1258-1268.

Schapire, R. E., & Singer, Y. (1998). Improved boosting algorithms using confidence-rated predictions. *Machine Learning, 37*(3), 297-336.

Westerdale, T. H. (2001). Local reinforcement and recombination in classifier systems. *Evolutionary Computation, 9*(3), 259-281.

Wilson, S. W. (2001). Mining oblique data with XCS. *Lecture Notes in Artificial Intelligence, 1996,* 158-176.

KEY TERMS

Classification: The central problem in (supervised) data mining. Given a training data set, classification algorithms provide predictions for new data based on predictive rules and other types of models.

Classifier System: A rich class of evolutionary computation algorithms building on the idea of evolving a population of predictive (or behavioral) rules under the enforcement of certain competition and cooperation processes. Note that classifier systems can also be understood as systems capable of performing classification. Not all CSs in the sense meant here qualify as classifier systems in the broader sense, but a variety of CS algorithms concerned with classification do.

Ensemble-Based Methods: A general technique that seeks to profit from the fact that multiple rule generation followed by prediction averaging reduces test error.

Evolutionary Computation: The solution approach guided by artificial evolution, which begins with random populations (of solution models), then iteratively applies algorithms of various kinds to find the best or fittest models.

Fitness Landscape: Optimization space due to the characteristics of the fitness measure used to define the evolutionary computation process.

Predictive Rules: Standard *if-then* rules with the consequent expressing some form of prediction about the output variable.

Rule Mining: A computer-intensive task whereby data sets are extensively probed for useful predictive rules.

Test Error: Learning systems should be evaluated with regard to their true error rate, which in practice is approximated by the error rate on test data, or test error.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 487-491, copyright 2005 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

On Explanation-Oriented Data Mining

Yiyu Yao
University of Regina, Canada

Yan Zhao
University of Regina, Canada

INTRODUCTION

The objective of data mining is to discover new and useful knowledge, in order to gain a better understanding of nature. This in fact is the goal of scientists when carrying out scientific research, independent in their various disciplines. This goal-oriented view enables us to re-examine data mining in a wider context of scientific research. The consequence after the immediate comparison between scientific research and data mining is that, an explanation discovery and evaluation task is added to the existing data mining framework. In this chapter, we elaborate the basic concerns and methods of explanation discovery and evaluation. Explanation-oriented association mining is employed as a concrete example to show the whole framework.

BACKGROUND

Scientific research and data mining have much in common in terms of their goals, tasks, processes and methodologies. As a recently emerged multi-disciplinary study, data mining and knowledge discovery can benefit from the long established studies of scientific research and investigation (Martella *et al*., 1999). By viewing data mining in a wider context of scientific research, we can obtain insights into the necessities and benefits of explanation discovery. The model of explanation-oriented data mining is a recent result from such an investigation (Yao *et al*., 2003). The basic idea of explanation-oriented data mining has drawn attentions from many researchers (Lin & Chalupsky, 2004; Yao, 2003) ever since the introduction of it.

Common Goals of Scientific Research and Data Mining

Scientific research is affected by the perceptions and the purposes of science. Martella *et al*. (1999) summarized the main purposes of science, namely, to describe and predict, to improve or manipulate the world around us, and to explain our world. The results of the scientific research process provide a description of an event or a phenomenon. The knowledge obtained from research helps us to make predictions about what will happen in the future. Research findings are useful for us to make an improvement in the subject matter. Research findings can be used to determine the best or the most effective interventions that will bring about desirable changes. Finally, scientists develop models and theories to explain why a phenomenon occurs.

Goals similar to those of scientific research have been discussed by many researchers in data mining. For example, Fayyad *et al*. (1996) identified two high-level goals of data mining as prediction and description. Prediction involves the use of some variables to predict the values of some other variables, and description focuses on patterns that describe the data. Ling *et al*. (2002) studied the issue of manipulation and action based on the discovered knowledge. Yao *et al*. (2003) introduced the notion of explanation-oriented data mining, which focuses on constructing models for the explanation of data mining results.

Common Processes of Scientific Research and Data Mining

The process of scientific research includes idea generation, problem definition, procedure design and planning, observation and experimentation, data analysis, result interpretation, and communication. It is possible to combine several phases into one, or to divide one phase into more detailed steps. The division between phases is not a clear cut. The research process does not follow a rigid sequencing of the phases. Iteration of different phrases may be necessary (Graziano & Raulin, 2000, Martella *et al*., 1999).

Many researchers have proposed and studied models of data mining processes (Fayyad *et al*.,

1996; Mannila, 1997; Yao *et al*., 2003; Zhong *et al*., 2001). The model that adds the explanation facility to the commonly used models is recently proposed by Yao *et al*. The process thus is composed by: date preprocessing, data transformation, pattern discovery and evaluation, explanation discovery and evaluation, and pattern representation. Like the research process, the process of data mining also is an iterative process and there is no clear cut between different phases. In fact, Zhong, *et al*. (2001) argued that it should be a dynamically organized process. The whole framework is illustrated in Figure 1.

There is a parallel correspondence between the processes of scientific research and data mining. The main difference lies in the subjects that perform the tasks. Research is carried out by scientists, and data mining is done by computer systems. In particular, data mining may be viewed as a study of domain-independent research methods with emphasis on data analysis. The higher and more abstract level of comparisons of, and connections between, scientific research and data mining may be further studied in levels that are more concrete. There are bi-directional benefits. The experiences and results from the studies of research methods can be applied to data mining problems; the data mining algorithms can be used to support scientific research.

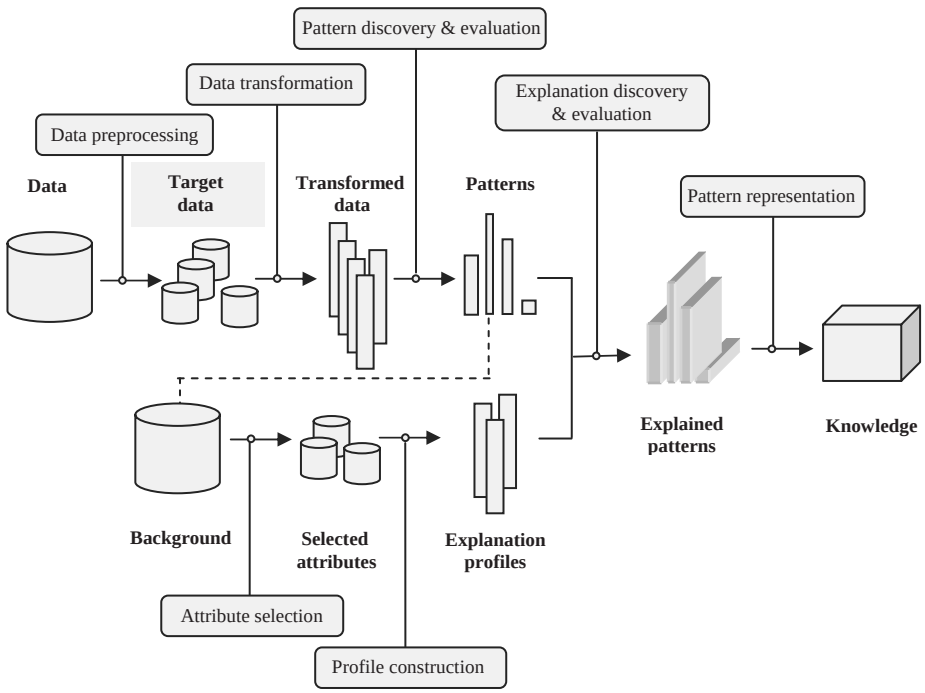
MAIN THRUST OF THE CHAPTER

Explanations of data mining address several important questions. What needs to be explained? How to explain the discovered knowledge? Moreover, is an explanation correct and complete? By answering these questions, one can better understand explanation-oriented data mining. In this section, the ideas and processes of explanation profile construction, explanation discovery and explanation evaluation are demonstrated by explanation-oriented association mining.

Basic Issues

- Explanation-oriented data mining explains and interprets the knowledge discovered from data. Knowledge can be discovered by unsupervised learning methods. Unsupervised learning studies how systems can learn to represent, summarize, and organize the data in a way that reflects the internal structure (namely, a pattern) of the

Figure 1. A framework of explanation-oriented data mining



overall collection. This process does not explain the patterns, but describes them. The primary unsupervised techniques include clustering mining, belief (usually Bayesian) networks learning, and association mining. The criteria for choosing the pattern to be explained are directly related to pattern evaluation step of data mining.

- Explanation-oriented data mining needs the background knowledge to infer features that can possibly explain a discovered pattern.
  The theory of explanation can be deeply impacted by considerations from many branches of inquiry: physics, chemistry, meteorology, human culture, logic, psychology, and the methodology of science above all. In data mining, explanation can be made at a shallow, syntactic level based on statistical information, or at a deep, semantic level based on domain knowledge. The required information and knowledge for explanation may not necessarily be inside the original dataset. A user needs to collect additional information for explanation discovery according to his/her background knowledge. It is argued that the power of explanations involves the power of insight and anticipation. One collects certain features based on the underlying hypothesis that they may provide explanations of the discovered pattern. That something is unexplainable may simply be an expression of the inability to discover an explanation of a desired sort. The process of selecting the relevant and explanatory features may be subjective, and trial-and-error. In general, the better our background knowledge, the more accurate the inferred explanations are likely to be.
- Explanation-oriented data mining explains inductively. That is, it draws an inference from a set of acquired training instances, and justifies or predicts the instances one might observe in the future.
  Supervised learning methods can be applied for the explanation discovery. The goal of supervised learning is to find a model that will correctly associate the input patterns with the classes. In real world applications, supervised learning models are extremely useful analytic techniques. The widely used supervised learning methods include decision tree learning, rule-based learning, and decision graph learning. The learned results are represented as either a tree, or a set of if-then rules.
- The constructed explanations give some evidence of under what conditions (within the background knowledge) the discovered pattern is most likely to happen, or how the background knowledge is related to the pattern.
  The role of explanation in data mining is positioned among proper description, relation and causality. Comprehensibility is the key factor in explanations. The accuracy of the constructed explanations relies on the amount of training examples. Explanation-oriented data mining performs poorly with insufficient data or poor presuppositions. Different background knowledge may infer different explanations. There is no reason to believe that only one unique explanation exists. One can use statistical measures and domain knowledge to evaluate different explanations.

A Concrete Example: Explanation-Oriented Association Mining

Explanations, also expressed as conditions, can provide additional semantics to a standard association. For example, by adding time, place, and/or customer profiles as conditions, one can identify when, where, and/or to whom an association occurs.

Explanation Profile Construction

The approach of explanation-oriented association mining combines unsupervised and supervised learning methods. It is to discover an association first, and then to explain the association. Conceptually, this method consists of two data tables. One table is used to learn associated patterns. An unsupervised learning algorithm, like the Apriori algorithm (Agrawal *et al*., 1993), can be applied to discover frequent associations. To discover other types of associations, different algorithms can be applied. By picking one interested associated pattern, we can assign objects as positive instances if they satisfy the desired associated pattern, and negative instances, otherwise. In the other table, a set of explanation profiles, with respect to the interested associated pattern, are collected according to the user's background knowledge and inferences. This table is used to search for explanations of the selected associated pattern. We can construct an explanation

table by combining the decisions label and explanation profiles as the schema that describes all the objects in the first table.

Explanation Discovery

Conditions that explain the associated pattern are searched by using a supervised learning algorithm in the constructed explanation table. A classical supervised learning algorithm such as ID3 (Quinlan, 1983), C4.5 (Quinlan, 1993), PRISM (Cendrowska, 1987), and many more (Brodie & Dejong, 2001; Han & Kamber, 2000; Mitchell, 1999) may be used to discover explanations. The Apriori-ID3 algorithm, which can be regarded as an example of explanation-oriented association mining method, is described below.

Explanation Evaluation

Once explanations are generated, it is necessary to evaluate them. For explanation-oriented association mining, we want to compare a conditional association (explained association) with its unconditional counterpart, as well as to compare different conditions.

Let T be a transaction table. Suppose for a desired pattern ϕ generated by an unsupervised learning algorithm from T, E is a constructed explanation profile table associated with ϕ. Suppose there is a set K of conditions (explanations) discovered by a supervised learning algorithm from E, and $\lambda \in K$ is one explanation. Two points are noted: first, the set K of explanations can be different according to various explanation profile tables, or various supervised learning algorithms. Second, not all explanations in K are equally interesting. Different conditions may have different degrees of interestingness.

Suppose ε is a quantitative measure used to evaluate plausible explanations, which can be the support measure for an undirected association, the confidence or coverage measure for a one-way association, or the similarity measure for a two-way association (refer to Yao & Zhong, 1999). A condition $\lambda \in K$ provides an explanation of a discovered pattern ϕ if $\varepsilon(\phi \mid \lambda) > \varepsilon(\phi)$. One can further evaluate explanations quantitatively based on several measures, such as absolute difference (*AD*), relative difference (*RD*) and ratio of change (*RC*):

$$AD(\phi \mid \lambda) = \varepsilon(\phi \mid \lambda) - \varepsilon(\phi),$$

$$RD(\phi \mid \lambda) = \frac{\varepsilon(\phi \mid \lambda) - \varepsilon(\phi)}{\varepsilon(\phi)},$$

$$RC(\phi \mid \lambda) = \frac{\varepsilon(\phi \mid \lambda) - \varepsilon(\phi)}{1 - \varepsilon(\phi)}.$$

The absolute difference represents the disparity between the pattern and the pattern under the condition. For a positive value, one may say that the condition supports ϕ, for a negative value, one may say that the condition rejects ϕ. The relative difference is the ratio of absolute difference to the value of the unconditional pattern. The ratio of change compares the actual change and the maximum potential change. Normally, a user is not interested in a pattern that cannot be explained. However, this situation might stimulate him/her to infer a different set of explanation profiles in order to construct a new set of explanations of the pattern.

Generality is the measure to quantify the size of a condition with respect to the whole data. When the generality of conditions is essential, a compound measure should be applied. For example, one may be interested

The Apriori-ID3 algorithm
Input: A transaction table and a related explanation profile table.
Output: Associations and explained associations.

1. Use the Apriori algorithm to generate a set of frequent associations in the transaction table. For each association $\phi \wedge \varphi$ in the set, *support*($\phi \wedge \varphi$) ≥ *minsup*, and *confidence*($\phi \Rightarrow \varphi$) ≥ *minconf*.
2. If the association $\phi \wedge \varphi$ is considered interesting (with respect to the user feedback or interestingness measures), then
   a. Introduce a binary attribute named *Decision*. Given a transaction, its value on *Decision* is "+" if it satisfies $\phi \wedge \varphi$ in the original transaction table, otherwise its value is "-".
   b. Construct an information table by using the attribute *Decision* and explanation profiles. The new table is called an *explanation table*.
   c. By treating *Decision* as the target class, apply the ID3 algorithm to derive classification rules of the form $\lambda \Rightarrow$ (*Decision* = "+"). The condition λ is a formula discovered in the explanation table, which states that under λ the association $\phi \wedge \varphi$ occurs.
   d. Evaluate the constructed explanation(s).

in discovering an accurate explanation with a high ratio of change and a high generality. However, it often happens that an explanation has a high generality but a low *RC* value, while another explanation has a low generality but a high *RC* value. A trade-off between these two explanations does not necessarily exist.

A good explanation system must be able to rank the discovered explanations and to reject bad explanations. It should be realized that evaluation is a difficult process because so many different kinds of knowledge can come into play. In many cases, one must rely on domain experts to reject uninteresting explanations.

Applications of Explanation-Oriented Data Mining

The idea of explanation-oriented association mining can be illustrated by a project called *WebAnalysis*, which attempts to explain Web browsing behaviour based on users' features (Zhao, 2003).

We use Active Server Pages (ASP) to create a set of Web pages. Ten popular categories of Web sites ranked by *PC Magazine* in 2001 are included. The association rule mining in the log file shows that many people who are interested in finance also are interested in business. We want to explain this frequent association by answering a more general question: "do users' profiles affect their browsing behaviours?" We thus recall users' personal information such as age, gender and occupation, which have been collected and stored in another log file, originally for security purposes. By combining the user profiles with the decisions on the association, we obtain some plausible explanations like: business-related readers in the age group of 30-39 most likely have this browsing habit; business-related female readers also are interested in these two categories.

Suppose we are not interested in users' demographics information, instead, we collect some other information such as users' access time and duration, operating system, browser and region. We may have different plausible explanations based on this different expectation.

An ontology-based clustering project, called ONTOCLUST, can be borrowed to embody the explanation-oriented clustering analysis (Wang & Hamilton, 2005). The project demonstrates the usage of spatial ontology for explaining the population distribution and clustering. The experiment shows that the resulting Canadian population clusters are matched with the locations of major cities and geographical area in the ontology, and then we can explain the clustering results.

FUTURE TRENDS

Considerable research remains to be done for explanation discovery and evaluation. In this chapter, rule-based explanation is discovered by inductive supervised learning algorithms. Alternatively, case-based explanations need to be addressed too. Based on the case-based explanation, a pattern is explained if an actual prior case is presented to provide compelling support. One of the perceived benefits of case-based explanation is that the rule generation effort is saved. Instead, some similarity functions need to be studied to score the distance between the descriptions of the newly discovered pattern and an existing one, and retrieve the most similar case as an explanation.

The discovered explanations of the selected pattern provide conclusive evidence for the new instances. In other words, the new instances can be explained and implied by the explanations. This is normally true when the explanations are sound and complete. However, sometimes, the discovered explanations cannot guarantee that a certain instance perfectly fit it. Even worse, a new dataset as a whole may show a change or a confliction with the learned explanations. This is because the explanations may be context-dependent on certain spatial and/or temporal interval. To consolidate the explanations we have discovered, a spatial-temporal reasoning model needs to be introduced to show the trend and evolution of the pattern to be explained.

CONCLUSION

Explanation-oriented data mining offers a new point of view. It closely relates scientific research and data mining, which have bi-directional benefits. The idea of explanation-oriented mining may have a significant impact on the understanding of data mining and effective applications of data mining results.

REFERENCES

Agrawal, R., Imielinski, T. & Swami, A. (1993). Mining association rules between sets of items in large

databases. *Proceedings of ACM Special Interest Group on Management of Data' 1993*, 207-216.

Brodie, M. & Dejong, G. (2001). Iterated phantom induction: A knowledge-based approach to learning control. *Machine Learning, 45(1)*, 45-76.

Cendrowska, J. (1987). PRISM: An algorithm for inducing modular rules. *International Journal of Man-Machine Studies, 27*, 349-370.

Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P. & Uthurusamy, R. (Eds.) (1996). *Advances in Knowledge Discovery and Data Mining*, AAAI/MIT Press.

Graziano, A.M & Raulin, M.L. (2000). *Research Methods: A Process of Inquiry*, 4th edition, Allyn & Bacon, Boston.

Han, J. & Kamber, M. (2000). *Data Mining: Concept and Techniques*, Morgan Kaufmann Publisher.

Lin, S. & Chalupsky, H. (2004). Issues of verification for unsupervised discovery systems. *Proceedings of KDD'2004 Workshop on Link Discovery.*

Ling, C.X., Chen, T., Yang, Q. & Cheng, J. (2002). Mining optimal actions for profitable CRM. *Proceedings of International Conference on Data Mining' 2002*, 767-770.

Martella, R.C., Nelson, R. & Marchand-Martella, N.E. (1999). *Research Methods: Learning to Become a Critical Research Consumer*, Allyn & Bacon, Boston.

Mannila, H. (1997). Methods and problems in data mining. *Proceedings of International Conference on Database Theory' 1997*, 41-55.

Mitchell, T. (1999). Machine learning and data mining. *Communications of the ACM, 42(11)*, 30-36.

Quinlan, J.R. (1983). Learning efficient classification procedures. In J.S. Michalski, J.G. Carbonell & T.M. Mirchell (Eds.), *Machine Learning: An Artificial Intelligence Approach, 1*, Morgan Kaufmann, Palo Alto, CA, 463-482.

Quinlan, J.R. (1993). *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publisher.

Wang, X. & Hamilton, H.J. (2005). Towards an ontology-based spatial clustering framework, *Proceedings of Conference of the Canadian Society for Computational Studies of Intelligence*, 205-216.

Yao, J.T. (2003). Sensitivity analysis for data mining. *Proceedings of Fuzzy Information Processing Society*, 272- 277.

Yao, Y.Y. (2003). A framework for web-based research support systems. *Proceedings of the Twenty-seventh Annual International Computer Software and Applications Conference*, 601-606.

Yao, Y.Y., Zhao, Y. & Maguire, R.B. (2003). Explanation-oriented association mining using rough set theory. *Proceedings of Rough Sets, Fuzzy Sets and Granular Computing*, 165-172.

Yao, Y.Y. & Zhong, N. (1999). An analysis of quantitative measures associated with rules. *Proceedings Pacific-Asia Conference on Knowledge Discovery and Data Mining' 1999*, 479-488.

Zhao, Y. (2003) *Explanation-oriented Data Mining*. Master Thesis. University of Regina.

Zhong, N., Liu, C. & Ohsuga, S. (2001). Dynamically organizing KDD processes. *International Journal of Pattern Recognition and Artificial Intelligence, 15*, 451-473.

KEY TERMS

Absolute Difference: A measure that represents the difference between an association and a conditional association based on a given measure. The condition provides a plausible explanation.

Explanation-Oriented Data Mining: A general framework includes data pre-processing, data transformation, pattern discovery and evaluation, explanation discovery and evaluation, and pattern presentation. This framework is consistent with the general model of scientific research processes.

Generality: A measure that quantifies the coverage of an explanation in the whole dataset.

Goals of Scientific Research: The purposes of science are to describe and predict, to improve or manipulate the world around us, and to explain our world. One goal of scientific research is to discover

new and useful knowledge for the purpose of science. As a specific research field, data mining shares this common goal, and may be considered as a research support system.

Method of Explanation-Oriented Data Mining: The method consists of two main steps and uses two data tables. One table is used to learn a pattern. The other table, an explanation profile table, is constructed and used for explaining one desired pattern. In the first step, an unsupervised learning algorithm is used to discover a pattern of interest. In the second step, by treating objects satisfying the pattern as positive instances, and treating the rest as negative instances, one can search for conditions that explain the pattern by a supervised learning algorithm.

Ratio of Change: A ratio of actual change (absolute difference) to the maximum potential change.

Relative Difference: A measure that represents the difference between an association and a conditional association relative to the association based on a given measure.

Scientific Research Processes: A general model consists of the following phases: idea generation, problem definition, procedure design/planning, observation/experimentation, data analysis, results interpretation, and communication. It is possible to combine several phases, or to divide one phase into more detailed steps. The division between phases is not a clear cut. Iteration of different phrases may be necessary.

Extending a Conceptual Multidimensional Model for Representing Spatial Data

Elzbieta Malinowski
Universidad de Costa Rica, Costa Rica

Esteban Zimányi
Université Libre de Bruxelles, Belgium

INTRODUCTION

Data warehouses keep large amounts of historical data in order to help users at different management levels to make more effective decisions. Conventional data warehouses are designed based on a multidimensional view of data. They are usually represented as *star* or *snowflake schemas* that contain relational tables called fact and dimension tables. A *fact table* expresses the focus of analysis (e.g., analysis of sales) and contains numeric data called *measures* (e.g., quantity). Measures can be analyzed according to different analysis criteria or *dimensions* (e.g., by product). Dimensions include attributes that can form *hierarchies* (e.g., product-category). Data in a data warehouse can be dynamically manipulated using on-line analysis processing (OLAP) systems. In particular, these systems allow automatic measure aggregations while traversing hierarchies. For example, the roll-up operation transforms detailed measures into aggregated data (e.g., daily into monthly sales) while the drill-down operation does the contrary.

Data warehouses typically include a location dimension, e.g., store or client address. This dimension is usually represented in an alphanumeric format. However, the advantages of using spatial data in the analysis process are well known since visualizing data in space allows users to reveal patterns that are difficult to discover otherwise. Spatial databases have been used for several decades for storing and managing spatial data. This kind of data typically represents geographical objects, i.e., objects located on the Earth's surface (such as mountains, cities) or geographic phenomena (such as temperature, altitude). Due to technological advances, the amount of available spatial data is growing considerably, e.g., satellite images, and location data from remote sensing systems, such as Global Positioning Systems (GPS). Spatial databases are typically used for daily business manipulations, e.g., to find a specific place from the current position given by a GPS. However, spatial databases are not well suited for supporting the decision-making process (Bédard, Rivest, & Proulx, 2007), e.g., to find the best location for a new store. Therefore, the field of spatial data warehouses emerged as a response to the necessity of analyzing high volumes of spatial data.

Since applications including spatial data are usually complex, they should be modeled at a conceptual level taking into account users' requirements and leaving out complex implementation details. The advantages of using conceptual models for database design are well known. In conventional data warehouses, a multidimensional model is commonly used for expressing users' requirements and for facilitating the subsequent implementation; however, in spatial data warehouses this model is seldom used. Further, existing conceptual models for spatial databases are not adequate for multidimensional modeling since they do not include the concepts of dimensions, hierarchies, and measures.

BACKGROUND

Only a few conceptual models for spatial data warehouse applications have been proposed in the literature (Jensen, Klygis, Pedersen, & Timko, 2004; Timko & Pedersen, 2004; Pestana, Mira da Silva, & Bédard, 2005; Ahmed & Miquel, 2005; Bimonte, Tchounikine, & Miquel, 2005). Some of these models include the concepts presented in Malinowski and Zimányi (2004) and Malinowski and Zimányi (2005), to which we will refer in the next section; other models extend non-spatial multidimensional models with different aspects, such as imprecision (Jensen *et al.*, 2004), location-based data (Timko & Pedersen, 2004), or continuous phenomena such as temperature or elevation (Ahmed & Miquel, 2005).

Other authors consider spatial dimensions and spatial measures (Stefanovic, Han, & Koperski, 2000; Rivest, Bédard, & Marchand, 2001; Fidalgo, Times, Silva, & Souza, 2004); however, their models are mainly based on the star and snowflake representations and have some restrictions, as we will see in the next section.

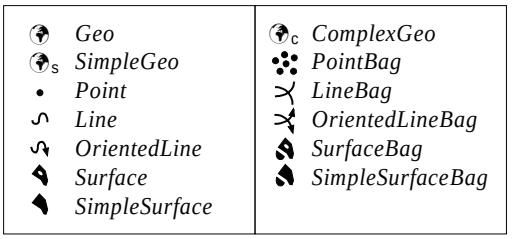
We advocate that it is necessary to have a conceptual multidimensional model that provides organized spatial data warehouse representation (Bédard, Merrett, & Han, 2001) facilitating spatial on-line analytical processing (Shekhar & Chalwa, 2003; Bédard *et al.*, 2007), spatial data mining (Miller & Han, 2001), and spatial statistical analysis. This model should be able to represent multidimensional elements, i.e., dimensions, hierarchies, facts, and measures, but also provide spatial support.

Spatial objects correspond to real-world entities for which the application needs to keep their spatial characteristics. Spatial objects consist of a *thematic* (or descriptive) component and a *spatial* component. The thematic component is represented using traditional DBMS data types, such as integer, string, and date. The spatial component includes its geometry, which can be of type point, line, surface, or a collection of these types. Spatial objects relate to each other with topological relationships. Different topological relationships have been defined (Egenhofer, 1993). They allow, e.g., determining whether two counties touches (i.e., share a common border), whether a highway crosses a county, or whether a city is inside a county.

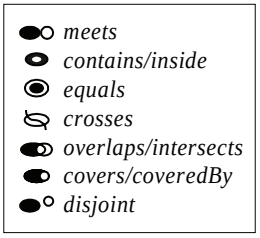
Pictograms are typically used for representing spatial objects and topological relationships in conceptual models. For example, the conceptual spatio-temporal model MADS (Parent *et al.* 2006) uses the pictograms shown in Figure 1.

The inclusion of spatial support in a conceptual multidimensional model should consider different aspects not present in conventional multidimensional models, such as the topological relationships existing between the different elements of the multidimensional model or aggregations of spatial measures, among others. While some of these aspects are briefly mentioned in the literature, e.g., spatial aggregations (Pedersen & Tryfona, 2001), others are neglected, e.g., the influence on aggregation procedures of the topological relationships between spatial objects forming hierarchies.

Figure 1. Pictograms for a) spatial data types and b) topological relationships



(a)



(b)

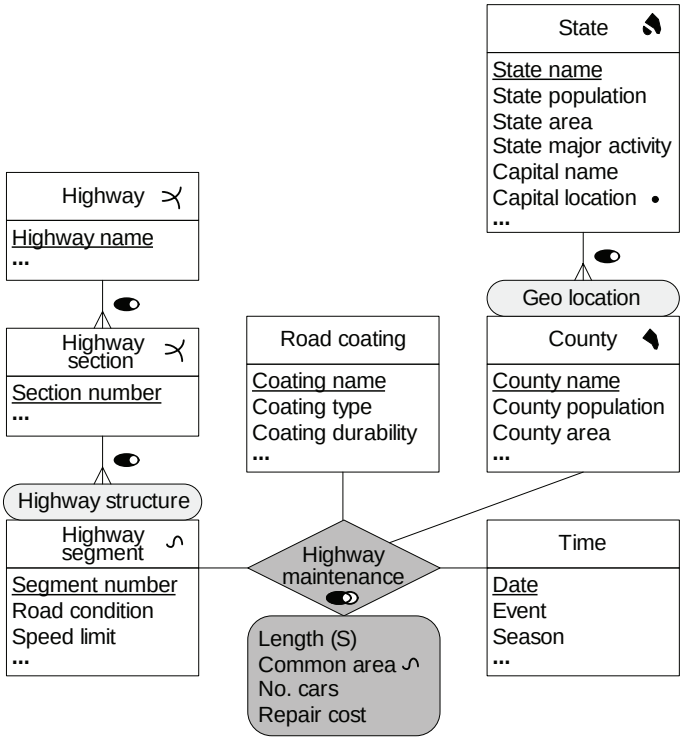
MAIN FOCUS

The MultiDim model (Malinowski & Zimányi, 2008a, 2008b) is a conceptual multidimensional model that allows designers to represent fact relationships, measures, dimensions, and hierarchies. It was extended by the inclusion of spatial support in the different elements of the model (Malinowski & Zimányi, 2004; Malinowski & Zimányi, 2005). We briefly present next our model.

A Conceptual Multidimensional Model for Spatial Data Warehouses

To describe the MultiDim model, we use an example concerning the analysis of highway maintenance costs. Highways are divided into highways sections, which at their turn are divided into highway segments. For each segment, the information about the number of cars and repairing cost during different periods of time is available. Since the maintenance of highway segments is the responsibility of counties through which the highway passes, the analysis should consider the administrative division of the territory, i.e., county and state. The analysis should also help to reveal how the different

Figure 2. An example of a multidimensional schema with spatial elements



types of road coating influence the maintenance costs. The multidimensional schema that represents these requirements is shown in Figure 2. To understand the constructs of the MutiDim model, we ignore for the moment the spatial support, i.e., the symbols for the geometries and the topological relationships.

A *dimension* is an abstract concept for grouping data that shares a common semantic meaning within the domain being modeled. It represents either a level or one or more hierarchies. *Levels* correspond to entity types in the entity-relationship model; they represent a set of instances, called *members*, having common characteristics. For example, Road Coating in Figure 2 is a one-level dimension.

Hierarchies are required for establishing meaningful paths for the roll-up and drill-down operations. A hierarchy contains several related levels, such as the County and State levels in Figure 2. They can express different structures according to an *analysis criterion*, e.g., geographical location. We use the criterion name to differentiate them, such as Geo location, or Highway structure in Figure 2.

Given two related levels of a hierarchy, one of them is called *child* and the other *parent* depending on whether they include more detailed or more general data, respectively. In Figure 2, Highway segment is a child level while Highway section is a parent level. A level of a hierarchy that does not have a child level is called *leaf* (e.g., Highway segment); the level that does not have a parent level is called *root* (e.g., Highway).

The relationships between child and parent levels are characterized by *cardinalities*. They indicate the minimum and the maximum numbers of members in one level that can be related to a member in another level. In Figure 2, the cardinality between the County and State levels is many-to-one indicating that a county can belong to only one state and a state can include many counties. Different cardinalities may exist between levels leading to different types of hierarchies (Malinowski & Zimányi, 2008a, 2008b).

Levels contain one or several *key attributes* (underlined in Figure 2) and may also have other *descriptive* attributes. Key attributes indicate how child members are grouped into parent members for the roll-up operation. For example, in Figure 2 since State name is the

key of the State level, counties will be grouped according to the state name to which they belong.

A *fact relationship* (e.g., Highway maintenance in Figure 2) represents an n-ary relationship between leaf levels. It expresses the focus of analysis and may contain attributes commonly called *measures* (e.g., Repair cost in the figure). They are used to perform quantitative analysis, such as to analyze the repairing cost during different periods of time.

Spatial Elements

The MultiDim model allows including spatial support for levels, attributes, fact relationships, and measures (Malinowski & Zimányi, 2004; Malinowski & Zimányi, 2005).

Spatial levels are levels for which the application needs to keep their spatial characteristics. This is captured by its geometry, which is represented using the pictograms shown in Figure 1 a). The schema in Figure 2 has five spatial levels: County, State, Highway segment, Highway section, and Highway. A level may have spatial attributes independently of the fact that it is spatial or not, e.g., in Figure 2 the spatial level State contains a spatial attribute Capital.

A *spatial dimension* (respectively, *spatial hierarchy*) is a dimension (respectively, a hierarchy) that includes at least one spatial level. Two linked spatial levels in a hierarchy are related through a topological relationship. These are represented using the pictograms of Figure 1 b). By default we suppose the *coveredBy* topological relationship, which indicates that the geometry of a child member is covered by the geometry of a parent member. For example, in Figure 2, the geometry of each county must be covered by the geometry of the corresponding state. However, in real-world situations different topological relationships can exist between spatial levels. It is important to consider these different topological relationships because they determine the complexity of the procedures for measure aggregation in roll-up operations (Malinowski & Zimányi, 2005).

A *spatial fact relationship* relates two or more spatial levels, e.g., in Figure 2 Highway maintenance relates the Highway segment and the County spatial levels. It may require the inclusion of a spatial predicate for spatial join operations. For example, in the figure an intersection topological relationship indicates that users focus their analysis on those highway segments that intersect counties. If this topological relationship is not included, users are interested in any topological relationships that may exist between them.

A (spatial) fact relationship may include measures, which may be spatial or thematic. *Thematic measures* are usual numeric measures as in conventional data warehouses, while *spatial measures* are represented by a geometry. Notice that thematic measures may be calculated using spatial operators, such as distance, area, etc. To indicate that a measure is calculated using spatial operators, we use the symbol (S). The schema in Figure 2 contains two measures. Length is a thematic measure (a number) calculated using spatial operators; it represents the length of the part of a highway segment that belongs to a county. Common area is a spatial measure representing the geometry of the common part. Measures require the specification of the function used for aggregations along the hierarchies. By default we use *sum* for numerical measures and *spatial union* for spatial measures.

Different Approaches for Spatial Data Warehouses

The MultiDim model extends current conceptual models for spatial data warehouses in several ways. It allows representing spatial and non-spatial elements in an orthogonal way; therefore users can choose the representation that better fits their analysis requirements. In particular we allow a non-spatial level (e.g., address represented by an alphanumeric data type) to roll-up to a spatial level (e.g., city represented by a surface). Further, we allow a dimension to be spatial even if it has only one spatial level, e.g., a State dimension that is spatial without any other geographical division. We also classify different kinds of spatial hierarchies existing in real-world situations (Malinowski & Zimányi, 2005) that are currently ignored in research related to spatial data warehouses. With respect to spatial measures, we based our approach on Stefanovic *et al.* (2000) and Rivest *et al.* (2001); however, we clearly separate the conceptual and the implementation aspects. Further, in our model a spatial measure can be related to non-spatial dimensions as can be seen in Figure 3.

For the schema in Figure 3 the user is interested in analyzing locations of accidents taking into account the different insurance categories (full coverage, partial coverage, etc.) and particular client data. The model includes a spatial measure representing the location of an accident. As already said above, a spatial function is

Figure 3. Schema for analysis of accidents with a spatial measure location

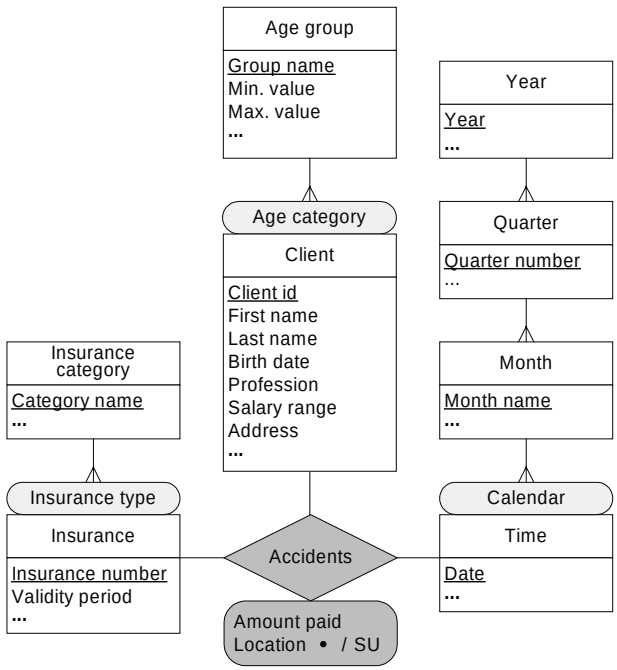
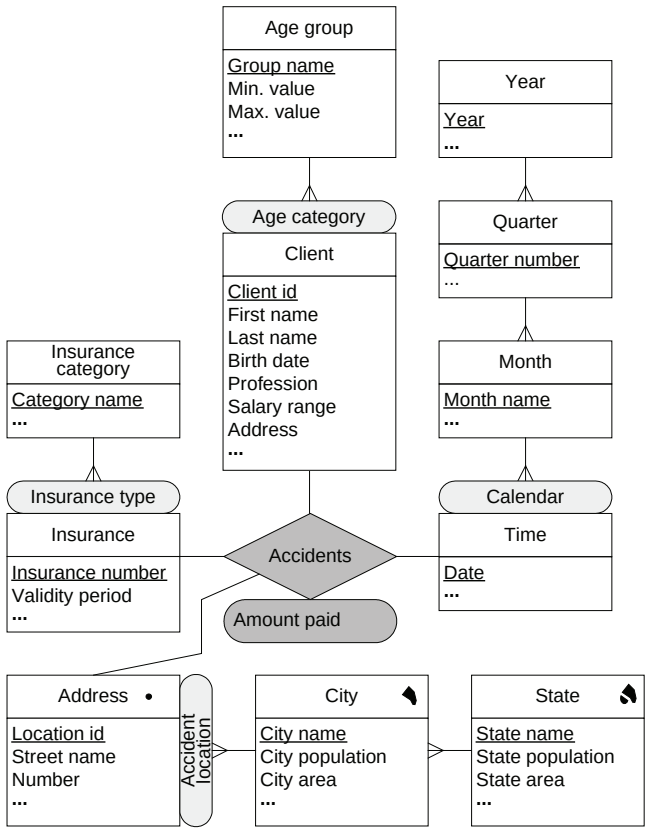


Figure 4. Spatial dimension: another variant of a schema for analysis of accidents



needed to aggregate spatial measures through hierarchies. By default the spatial union is used: when a user rolls-up to the Insurance category level, the locations corresponding to different categories will be aggregated and represented as a set of points. Other spatial operators can be also used, e.g., center of n points.

Other models (e.g., Fidalgo *et al.*, 2004) do not allow spatial measures and convert them into spatial dimensions. However, the resulting model corresponds to different analysis criteria and answers to different queries. For example, Figure 4 shows an alternative schema for the analysis of accidents that results from transforming the spatial measure Location of the Figure 3 into a spatial dimension Address.

For the schema in Figure 4 the focus of analysis has been changed to the amount of insurance paid according to different geographical locations. Therefore, using this schema, users can compare the amount of insurance paid in different geographic zones; however, they cannot aggregate locations (e.g., using spatial union) of accidents as can be done for the schema in Figure 3. As can be seen, although these models are similar, different analyses can be made when a location is handled as a spatial measure or as a spatial hierarchy. It is the designer's decision to determine which of these models better represents users' needs.

To show the feasibility of implementing our spatial multidimensional model, we present in Malinowski and Zimányi, (2007, 2008b) their mappings to the object-relational model and give examples of their implementation in Oracle 10g.

FUTURE TRENDS

Bédard *et al.*, (2007) developed a spatial OLAP tool that includes the roll-up and drill-down operations. However, it is necessary to extend these operations for different types of spatial hierarchies (Malinowski & Zimányi, 2005).

Another interesting research problem is the inclusion of spatial data represented as continuous fields, such as temperature, altitude, or soil cover. Although some solutions already exist (Ahmed & Miquel, 2005), additional research is required in different issues, e.g., spatial hierarchies composed by levels representing field data or spatial measures representing continuous phenomena and their aggregations.

Another issue is to cope with multiple representations of spatial data, i.e., allowing the same real-world object to have different geometries. Multiple representations are common in spatial databases. It is also an important aspect in the context of data warehouses since spatial data may be integrated from different source systems that use different geometries for the same spatial object. An additional difficulty arises when the levels composing a hierarchy can have multiple representations and one of them must be chosen during roll-up and drill-down operations.

CONCLUSION

In this paper, we referred to spatial data warehouses as a combination of conventional data warehouses and spatial databases. We presented different elements of a spatial multidimensional model, such as spatial levels, spatial hierarchies, spatial fact relationships, and spatial measures.

The spatial extension of the conceptual multidimensional model aims at improving the data analysis and design for spatial data warehouse and spatial OLAP applications by integrating spatial components in a multidimensional model. Being platform independent, it helps to establish a communication bridge between users and designers. It reduces the difficulties of modeling spatial applications, since decision-making users do not usually possess the expertise required by the software used for managing spatial data. Further, spatial OLAP tools developers can have a common vision of the different features that comprise a spatial multidimensional model and of the different roles that each element of this model plays. This can help to develop correct and efficient solutions for spatial data manipulations.

REFERENCES

Ahmed, T., & Miquel, M. (2005). Multidimensional structures dedicated to continuous spatio-temporal phenomena. *Proceedings of the 22nd British National Conference on Databases*, pp. 29-40. Lecture Notes in Computer Science, N° 3567. Springer.

Bédard, Y., Merrett, T., & Han, J. (2001). Fundamentals of spatial data warehousing for geographic knowledge

discovery. In J. Han & H. Miller (Eds.), *Geographic Data Mining and Knowledge Discovery*, pp. 53-73. Taylor & Francis.

Bédard, Y., Rivest, S., & Proulx, M. (2007). Spatial On-Line Analytical Processing (SOLAP): Concepts, Architectures and Solutions from a Geomatics Engineering Perspective. In R. Wrembel & Ch. Koncilia (Eds.), *Data Warehouses and OLAP: Concepts, Architectures and Solutions*, pp. 298-319. Idea Group Publishing.

Bimonte, S., Tchounikine, A., & Miquel, M. (2005). Towards a spatial multidimensional model. *Proceedings of the 8th ACM International Workshop on Data Warehousing and OLAP,* 39-46.

Egenhofer, M. (1993). A Model for detailed binary topological relationships, *Geomatica, 47*(3&4), 261-273.

Fidalgo, R., Times, V., Silva, J., & Souza, F. (2004). GeoDWFrame: A framework for guiding the design of geographical dimensional schemes. *Proceedings of the 6th International Conference on Data Warehousing and Knowledge Discovery,* pp. 26-37. Lecture Notes in Computer Science, N° 3181. Springer.

Jensen, C.S., Klygis, A., Pedersen, T., & Timko, I. (2004). Multidimensional Data Modeling for Location-Based Services. *VLDB Journal, 13*(1), pp. 1-21.

Malinowski, E. & Zimányi, E. (2004). Representing Spatiality in a Conceptual Multidimensional Model. *Proceedings of the 12th ACM Symposium on Advances in Geographic Information Systems,* pp. 12-21.

Malinowski, E. & Zimányi, E. (2005). Spatial Hierarchies and Topological Relationships in the Spatial MultiDimER model. *Proceedings of the 22nd British National Conference on Databases,* pp. 17-28. Lecture Notes in Computer Science, N° 3567. Springer.

Malinowski, E. & Zimányi, E. (2007). Spatial Hierarchies and Topological Relationships in the Spatial MultiDimER model. *GeoInformatica*, 11(4), 431-457.

Malinowski, E. & Zimányi, E. (2008a). Multidimensional Conceptual Models, *in this book.*

Malinowski, E. & Zimányi, E. (2008b). *Advanced Data Warehouse Design: From Conventional to Spatial and Temporal Applications.* Springer.

Pedersen, T.B. & Tryfona, N. (2001). Pre-aggregation in spatial data warehouses. *Proceedings of the 7th International Symposium on Advances in Spatial and Temporal Databases*, pp. 460-480. Lecture Notes in Computer Science, N° 2121. Springer.

Miller, H. & Han, J. (2001) *Geographic Data Mining and Knowledge Discovery.* Taylor & Francis.

Parent, Ch., Spaccapietra, S., & Zimányi, E. (2006). *Conceptual modeling for traditional and spatio-temporal applications: The MADS approach.* Springer.

Pestana, G., Mira da Silva, M., & Bédard, Y. (2005). Spatial OLAP modeling: An overview based on spatial objects changing over time. *Proceedings of the IEEE 3rd International Conference on Computational Cybernetics*, pp. 149-154.

Rivest, S., Bédard, Y., & Marchand, P. (2001). Toward better support for spatial decision making: defining the characteristics of spatial on-line analytical processing (SOLAP). *Geomatica, 55*(4), 539-555.

Shekhar, S., & Chawla, S. (2003). *Spatial Databases: A Tour*. Prentice Hall.

Stefanovic, N., Han, J., & Koperski, K. (2000). Object-based selective materialization for efficient implementation of spatial data cubes. *Transactions on Knowledge and Data Engineering, 12*(6), pp. 938-958.

Timko, I. & Pedersen, T. (2004). Capturing complex multidimensional data in location-based data warehouses. *Proceedings of the 12th ACM Symposium on Advances in Geographic Information Systems*, pp. 147-156.

KEY TERMS

Multidimensional Model: A model for representing the information requirements of analytical applications. It comprises facts, measures, dimensions, and hierarchies.

Spatial Data Warehouse: A data warehouse that includes spatial dimensions, spatial measures, or both, thus allowing spatial analysis.

Spatial Dimension: An abstract concept for grouping data that shares a common semantics within the

domain being modeled. It contains one or more spatial hierarchies.

Spatial Fact Relationship: An n-ary relationship between two or more spatial levels belonging to different spatial dimensions.

Spatial Hierarchy: One or several related levels where at least one of them is spatial.

Spatial Level: A type defining a set of attributes, one of them being the geometry, keeping track of the spatial extent and location of the instances, or members, of the level.

Spatial Measure: An attribute of a (spatial) fact relationship that can be represented by a geometry or calculated using spatial operators.

Facial Recognition

Rory A. Lewis
UNC-Charlotte, USA

Zbigniew W. Ras
University of North Carolina, Charlotte, USA

INTRODUCTION

Over the past decade Facial Recognition has become more cohesive and reliable than ever before. We begin with an analysis explaining why certain facial recognition methodologies examined under FERET, FRVT 2000, FRVT 2002, and FRVT 2006 have become stronger and why other approaches to facial recognition are losing traction. Second, we cluster the stronger approaches in terms of what approaches are mutually inclusive or exclusive to surrounding methodologies. Third, we discuss and compare emerging facial recognition technology in light of the aforementioned clusters. In conclusion, we suggest a road map that takes into consideration the final goals of each cluster, that given each clusters weakness, will make it easier to combine methodologies with surrounding clusters.

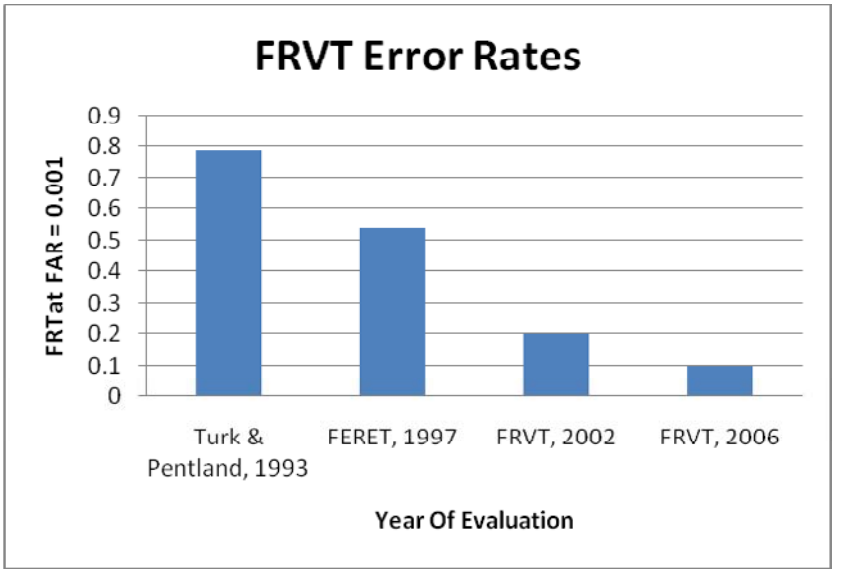
BACKGROUND

The National Institute of Standards and Technology (NIST) sponsored 2006 Face Recognition Vendor Test (FRVT) which is the most recent large scale independent synopsis of the state-of-the-art for face recognition systems. The previous tests in the series were the FERET, FRVT 2000, and FRVT 2002. The following organizations participated in the FRVT 2006 evaluation: Animetrics, Inc., Carnegie Mellon University, Cognitec Systems GmbH, Diamond Information Systems (DIS), Geometrix, Inc., Guardia, Identix, Inc., Neven Vision, New Jersey Institute of Technology (NJIT), Nivis, LLC, Old Dominion University, Panvista Limited, Peking University, Center for Information Science, PeopleSpot Inc., Rafael Armament Development Authority Ltd., SAGEM SA, Samsung Advanced Institute of Technology (SAIT), Tsinghua University, Tili Technology Limited, Toshiba Corporation, University of Houston, and Viisage. It should be noted that while the FRVT 2006 was conducted by the National Institute of Standards and Technology (NIST), it was jointly sponsored by five other U.S. Government agencies which share NIST's interest in measuring the improvements in face recognition technologies: Federal Bureau of Investigation, National Institute of Justice, National Institute of Standards and Technology, U.S. Department of Homeland Security, and the Transportation Security Administration.

The FRVT 2006 measured the progress of facial recognition systems including commercial systems that used Windows or Linux based algorithms. The sequestered data comprised a large standard dataset of "full frontal" pictures provided to NIST by the U.S. State Department using non-conforming pixel resolutions and lighting angles of 36,000 pictures of persons applying for non-immigrant visas at U.S. consulates in Mexico. The tests evaluated 4 dimensions of facial recognition: high resolution still imagery, 3D facial scans, multi-sample still facial imagery, and pre-processing algorithms that compensate for pose and illumination. The results of the best of the 13 groups that entered have improved remarkably; the best algorithms in the FRVT 2002 computed 20% false rejections compared to only 1% false rejections in the FRVT 2006 tests. However, some of the groups that entered FRVT 2006 had results no better than that of 2002. In the tests, the rejection was less palatable: 12% for the best algorithms which still it is better than the 29% rejection rate of the 2002 tests.

FRVT tests digress from the traditional facial recognition tests of the 1990's in two ways: First, speed was not the issue in the tests, some of the algorithms took hundreds of hours to find matches in the database. The correct identification (precision) is the issue. Secondly, rather than the traditional ID searches of comparing a face in the camera with every face in the database for a match, the FRVT tests comprised security verification: is the face of the person standing in front of the

Figure 1. The reduction in error rates of facial recognition algorithms



camera claiming to be Mr. Whomever indeed the Mr. Whomever whose picture is in the database?

THE STATE-OF-THE-ART OF THE 13 GROUPS IN FRVT 2006

The three well known facial recognition corporations, Google owned Neven Vision, Viisage Technology (owned by L-1 Identity Solutions), and Cognitec Systems of Germany performed the best. The two universities that excelled were the University of Houston and Tsinghua University in China. The four methodology clusters are (i) Support Vector Machines, (ii) Manifold/Eigenface, (iii) Principal Component Analysis with Modified Sum Square Error (PCA/SSE), and Pure Eigenface technology:

1. *Cognitec Systems* - Cognitec Systems FaceVACS-incorporates Support Vector Machines (SVM) to capture facial features. In the event of a positive match, the authorized person is granted access to a PC (Thalheim, 2002).
2. *Neven Vision* - In 2002, Neven's Eyematic team achieved high scores in the Face Recognition Vendor Test (Neven, 2004). Neven Vision incorporates Eigenface Technology. The Neven methodology trains an RBF network which constructs a full manifold representation in a universal Eigenspace from a single view of an arbitrary pose.
3. *L-1 Identity Solutions Inc* – L-1's performance ranked it near or at the top of every NIST test. This validated the functionality of the algorithms that drive L-1's facial and iris biometric solutions. L-1 was formed in 2006 when Viisage Technology Inc. bought Indentix Inc. The NIST evaluations of facial and iris technology covered algorithms submitted by Viisage and Indentix, both of which utilize variations of Eigenvalues, one called the principal component analysis (PCA)-based face recognition method and the other, the modified sum square error (SSE)-based distance technique.
4. *M A. Turk and A P. Pentland* – they wrote a paper that changed the facial recognition world - "*Face*

Recognition Using Eigenfaces" (Turk & Pentland, 1991) which won the "Most Influential Paper of the Decade" award from the IAPR MVA. In short, the new technology reconstructed a face by superimposing a set of "Eigenfaces". The similarity between the two facial images is determined based on the coefficient of the relevant Eigenfaces. The images consist of a set of linear eigenvectors wherein the operators are non-zero vectors which result in a scalar multiple of themselves (Pentland, 1993). It is the scalar multiple which is referred to as the Eigenvalue associated with a particular eigenvector (Pentland & Kirby, 1987). Turk assumes that any human face is merely a combination of parts of the Eigenfaces. One person's face may consist of 25% from face 1, 25% from face 2, 4% of face 3, n% of face X (Kirby, 1990).

The authors propose that even though the technology is improving, as exemplified by the FRVT 2006 test results, the common dominator for problematic recognition is based on faces with erroneous and varying light source conditions. Under this assumption, 3-D models were compared with the related 2-D images in the data set. All four of the aforementioned primary methodologies of facial recognition across the board, performed poorly in this context: for the controlled illumination experiments, performance of the very-high resolution dataset was better than the high-resolution dataset. Also, from comparing the photos of faces in the database to the angle of view of the person in 3-d/real life, an issue FRVT did not even consider, leads one to believe that light source variation on 3-d models compromises the knowledge discovery abilities of all the facial recognition algorithms at FRVT 2006.

Figure 2. Original picture set

Figure 3. RGB normalization

Figure 4. Grayscale and posturization

INNERTRON TECHNOLOGY

Major components of interactive visual data analysis and their functions that make knowledge extraction more effective are the current research theme in this field. Consider the 8 images of the subject photographed in Figure 2. These photos were taken at four different times over the period of 18 months under 4 varying sets of light sources. To the human brain – this is clearly the same person – without a doubt. However, after normalizing the light in RGB in each of the four sets, we yield the results displayed in Figure 3. Note that this normalization process is normalizing each group of four until the desired, empirical normalization matrix in RGB is achieved – as in the manner that a machine would render the normalization process. Note that in Figure 2 the bottom right hand figure is not darkened at all but after normalization in Figure 3 it appears darker. In Figure 4, a simple Grayscale and 6-level posturization is performed. In Figure 5, as in the FRVT 2006 tests, shadow is added to compensate for distance measures. It is evident that varying light conditions change the dynamics of a face in a far more complex manner than facial surgery could. This is evident when one looks at the 2-d images in terms of what is white and what is black, measuring and storing where they are black and white.

Innertron facial recognition technology proposes a solution to the core, common denominator problem of the FRVT 2006 tests as illustrated in Figures 2 through 5, where classical lighting and shadow normalization can corrupt lighting on some good images in order to compensate for bad lighting at the level of the median lighting parameters. The Lambertian model states that the image of a 3D object is subject to three factors, namely the surface normal, albedo, and lighting. These can be factorized into an intrinsic part of the surface normal and albedo, and an extrinsic part of lighting (Wang 2004).

Figure 5. Distance configuration

Figure 6. Difference Eigen

$$I(x,y) = \rho(x,y)n(x,y)^T s$$

where *p* is the surface texture (albedo) of the face, $\boldsymbol{n(x,y)^T}$ represents the 3-D rendering of the normal surface of the face, which will be the same for all objects of the class, and *s* is the position of the light source, which may vary arbitrarily. The quotient image $\boldsymbol{Q_y}$ of face ***y*** against face ***a*** is defined by

$$Q_y(u,v) = \frac{\rho_y(u,v)}{\rho_a(u,v)} = \frac{\rho_y(u,v)n(u,v)^T s_y}{\rho_a(u,v)n(u,v)^T s_y}$$
$$= \frac{I_y}{\rho_a(u,v)n(u,v)^T \sum_j x_j s_j}$$
$$= \frac{I_y}{\sum_{j=1}^{3} I_j x_j}$$

where ***u*** and ***v*** range over the image, $\boldsymbol{I_y}$ is an image of object ***y*** with the illumination direction $\boldsymbol{s_y}$, and $\boldsymbol{x_j}$ are combining coefficients estimated by Least Square based on training set (Shashua, 2001), (Raviv, 1999), and $\boldsymbol{I_1}$, $\boldsymbol{I_2}$, $\boldsymbol{I_3}$ are three non-collinearly illuminated images. This is shown in Figure 6 where we superimpose, using negative alpha values, face ***y*** onto face ***a*** with the higher level image additionally set to a basic Eigenvector exposing the differences of itself ***a*** and the image below it ***y***. This cancels out erroneous shadows, and only keeps constant data. The color of the images has changed from that of the original images in Figure 2. This is because of the effect the Eigenvector has on two samples from almost the same place and time. One can also see in Figure 7 that the large amount of shadows that were present in Figures 3 through 4 are almost gone. In Figure 7 and Figure 8 the innertron based strategy has normalized the face. *See* details in (Lewis, Ras 2005). In our approach, we use training database of human faces described by a set of innertron features to built classifiers for face recognition.

Figure 7. Calculate tilt

Figure 8. Normalize - grid

FUTURE TRENDS

There are many projects focused on creation of intelligent methods for solving many difficult high-level image feature extraction and data analysis problems in which both local and global properties as well as spatial relations are taken into consideration. Some of these new techniques work in pixel and compressed domain (avoiding the inverse transform in decompression), thus speeding up the whole process (Delac & Grgic, 2007), (Deniz et al., 2001), (Du et al., 2005). The development of these new high-level image features is essential for the successful construction of new classifiers for the face recognition. Knowledge discovery techniques are especially promising in providing tools to build classifiers for recognizing emotions associated with human faces (happy, sad, surprised, …..). But this area is still at the very beginning stage.

CONCLUSION

Overcoming the challenges of shadow elimination seems promising with the use of negative alpha values set to basic Eigenvectors particularly when used with the Lambertian methodology. Mining these images in a 2-D database is easier because the many levels of grey will be "normalized" to a specific set allowing a

more precise vector value for distance functions. Pure Eigenvector technology, also fails when huge amounts of "negatives" are superimposed onto one another, once again pointing to normalization as the key. Data in the innertron database need to be consistent in tilt and size of the face. This means that before a face is submitted to the database, it also needs tilt and facial size normalization.

REFERENCES

Delac, K., Grgic, M. (2007). Face Recognition, *I-Tech Education and Publishing*, Vienna, 558 pages.

Deniz, O., Castrillon, M., Hernández, M. (2001). Face recognition using independent component analysis and support vector machines, *Proceedings of Audio- and Video-Based Biometric Person Authentication: Third International Conference*, LNCS 2091, Springer.

Du, W., Inoue, K., Urahama, K. (2005). Dimensionality reduction for semi-supervised face recognition, *Fuzzy Systems and Knowledge Discovery (FSKD),* LNAI 3613-14, Springer, 1-10.

Kirby, M., Sirovich, L. (1990). Application of the Karhunen-Loeve procedure for the characterization of human faces, *IEEE Transactions on Pattern Analysis and Machine Intelligence 12(1),* 103 – 108.

Lewis, R.A., Ras, Z.W. (2005). New methodology of facial recognition, *Intelligent Information Processing and Web Mining*, Advances in Soft Computing, Proceedings of the IIS'2005 Symposium, Gdansk, Poland, Springer, 615-632.

Neven, H. (2004). *Neven Vision Machine Technology*, See: http://www.nevenvision.com/co neven.html

Pentland, L., Kirby, M. (1987). Low-dimensional procedure for the characterization of human faces, *Journal of the Optical Society of America 4*, 519 -524.

Pentland, A., Moghaddam, B., Starner, T., Oliyide, O., Turk, M. (1993). View-Based and modular eigenspaces for face recognition, *Technical Report 245*, MIT Media Lab.

Riklin-Raviv, T., Shashua, A. (1999). The Quotient image: class based recognition and synthesis under varying illumination, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 566 - 571

Shashua, A., Riklin-Raviv, T. (2001). The quotient image: class-based re-rendering and recognition with varying illuminations, *Transactions on Pattern Analysis and Machine Intelligence 23(2),* 129 - 139

Turk M., Pentland, A. (1991). Face recognition using eigenfaces, *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society Conference, 586-591

Wang, H., Li, S., Wang, Y. (2004). Face recognition under varying lighting conditions using self quotient image, *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, 819 – 824

KEY TERMS

Classifier: Compact description of a set of instances that have common behavioral and structural features (attributes and operations, respectively).

Eigenfaces: Set of eigenvectors used in the computer vision problem of human face recognition.

Eigenvalue: Scalar associated with a given linear transformation of a vector space and having the property that there is some nonzero vector which when multiplied by the scalar is equal to the vector obtained by letting the transformation operate on the vector.

Facial Recognition: Process which automatically identifies a person from a digital image or a video frame from a video source.

Knowledge Discovery: Process of automatically searching large volumes of data for patterns that can be considered as knowledge about the data.

Support Vector Machines (SVM): Set of related supervised learning methods used for classification and regression. They belong to a family of generalized linear classifiers. A special property of SVMs is that they simultaneously minimize the empirical classification error and maximize the geometric margin; hence they are also known as maximum margin classifiers.

F

Feature Extraction/Selection in High-Dimensional Spectral Data

Seoung Bum Kim
The University of Texas at Arlington, USA

INTRODUCTION

Development of advanced sensing technology has multiplied the volume of spectral data, which is one of the most common types of data encountered in many research fields that require advanced mathematical methods with highly efficient computation. Examples of the fields in which spectral data abound include near-infrared, mass spectroscopy, magnetic resonance imaging, and nuclear magnetic resonance spectroscopy.

The introduction of a variety of spectroscopic techniques makes it possible to investigate changes in composition in a spectrum and to quantify them without complex preparation of samples. However, a major limitation in the analysis of spectral data lies in the complexity of the signals generated by the presence of a large number of correlated features. Figure 1 displays a high-level diagram of the overall process of modeling and analyzing spectral data.

The collected spectra should be first preprocessed to ensure high quality data. Preprocessing steps generally include denoising, baseline correction, alignment, and normalization. Feature extraction/selection identifies the important features for prediction, and relevant models are constructed through the learning processes. The feedback path from the results of the validation step enables control and optimization of all previous steps. Explanatory analysis and visualization can provide initial guidelines that make the subsequent steps more efficient.

This chapter focuses on the feature extraction/selection step in the modeling and analysis of spectral data. Particularly, throughout the chapter, the properties of feature extraction/selection procedures are demonstrated with spectral data from high-resolution nuclear magnetic resonance spectroscopy, one of the widely used techniques for studying metabolomics.

BACKGROUND

Metabolomics is global analysis for the detection and recognition of metabolic changes in biological systems in response to pathophysiological stimuli and to the intake of toxins or nutrition (Nicholson et al., 2002). A variety of techniques, including electrophoresis, chromatography, mass spectroscopy, and nuclear magnetic resonance, are available for studying metabolomics. Among these techniques, proton nuclear magnetic resonance (^{1}H-NMR) has the advantages of high-resolution, minimal cost, and little sample preparation (Dunn & Ellis, 2005). Moreover, the tech-

Figure 1. Overall process for the modeling and analysis of high-dimensional spectra data

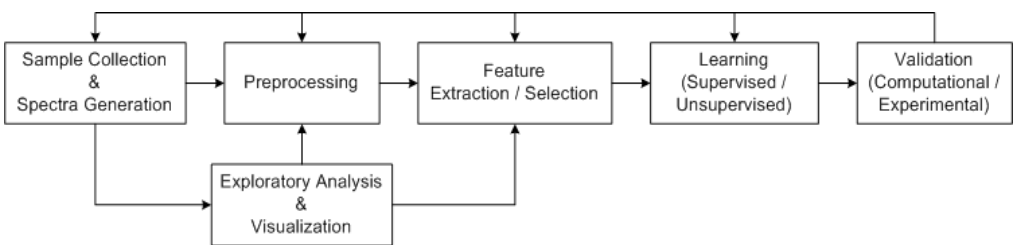
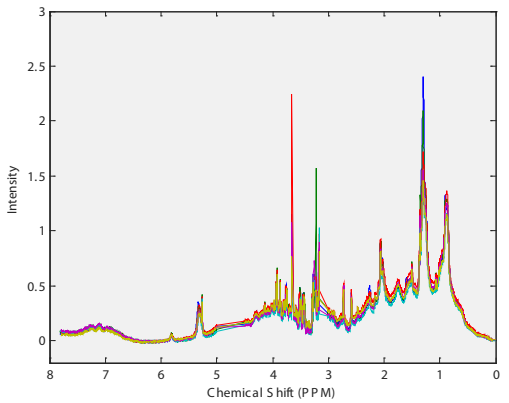


Figure 2. Multiple spectra generated by a 600MHz ^{1}H-NMR spectroscopy



nique generates high-throughput data, which permits simultaneous investigation of hundreds of metabolite features. Figure 2 shows a set of spectra generated by a 600MHz ^{1}H-NMR spectroscopy. The *x*-axis indicates the chemical shift within units in parts per million (ppm), and the *y*-axis indicates the intensity values corresponding to each chemical shift. Traditionally, chemical shifts in the *x*-axis are listed from largest to smallest. Analysis of high-resolution NMR spectra usually involves combinations of multiple samples, each with tens of thousands of correlated metabolite features with different scales.

This leads to a huge number of data points and a situation that challenges analytical and computational capabilities. A variety of multivariate statistical methods have been introduced to reduce the complexity of metabolic spectra and thus help identify meaningful patterns in high-resolution NMR spectra (Holmes & Antti, 2002). Principal components analysis (PCA) and clustering analysis are examples of unsupervised methods that have been widely used to facilitate the extraction of implicit patterns and elicit the natural groupings of the spectral dataset without prior information about the sample class (e.g., Beckonert et al., 2003). Supervised methods have been applied to classify metabolic profiles according to their various conditions (e.g., Holmes et al., 2001). The widely used supervised methods in metabolomics include Partial Least Squares (PLS) methods, *k*-nearest neighbors, and neural networks (Lindon et al., 2001).

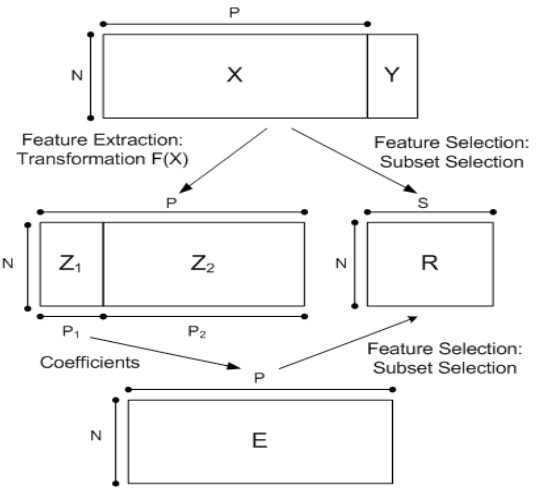
Although supervised and unsupervised methods have been successfully used for descriptive and predictive analyses in metabolomics, relatively few attempts have been made to identify the metabolite features that play an important role in discriminating between spectra among experimental conditions. Identifying important features in NMR spectra is challenging and poses the following problems that restrict the applicability of conventional methods. First, the number of features present usually greatly exceeds the number of samples (i.e., tall and wide data), which leads to ill-posed problems. Second, the features in a spectrum are correlated with each other, while many conventional multivariate statistical approaches assume the features are independent. Third, spectra comprise a number of local bumps and peaks with different scales.

MAIN FOCUS OF CHAPTER

Feature Extraction/Selection Process

It is important to distinguish between feature extraction and feature selection, although much of the literature

Figure 3. Overview of feature extraction, feature selection, and a combination of feature extraction and selection



fails to make a clear distinction between feature extraction and feature selection processes. Figure 3 displays a schematic diagram of the processes in feature extraction, feature selection, and a combination of feature extraction and feature selection.

Feature extraction techniques attempt to create new features based on transformations of the original features to extract the useful information for the model. Given the set of original features $\mathbf{X}_{(N\times P)}$, the main purpose of feature extraction is to find the function of $\mathbf{X}$, $\mathbf{Z}_{(N\times P1)} = F(\mathbf{X})$ to lower the original dimension. Generally, the first few dimensions are sufficient to account for the key information of the entire spectral data ($\mathbf{Z}_1$ in Figure 3). Widely used feature extraction methods include PCA, PLS, and Fisher discriminant analysis (FDA). Among these, PCA is an unsupervised feature extraction method because the process depends solely upon $\mathbf{X}$, while PLS and FDA are supervised methods because to extract features from X, they take into account the response variable. Recent study conducts an analysis and a comparison of several unsupervised linear feature selection methods in hyperspectral data (Jiménez-Rodríguez et al., 2007).

Feature selection techniques attempt to pick the subset of original features (with dimension S, where S « P, see Figure 3) that leads to the best prediction or classification. Feature selection methods have not been investigated as thoroughly as those for feature extraction because conventional feature selection methods cannot fully accommodate the correlation structure of features in a spectrum. This chapter presents two feature selection approaches that have the potential to cope with correlated features. These methods are based on a genetic algorithm and a multiple hypothesis testing procedure, respectively. Feature extraction is often the forerunner of feature selection. The coefficients that constitute the extracted features reflect the contribution of each feature and can be used to determine whether or not each feature is significant. These methods use loading vectors of PCA, the optimal discriminant direction of FDA, and index values of variable importance in projection in PLS.

Feature Extraction Techniques

Principal Component Analysis: PCA identifies a lower dimensional space that can explain most of the variability of the original dataset (Jollife, 2002). Let the random vector that has a sample of observations (n) for a set of p variables (i.e., $X^T=[X_1, X_2, \ldots, X_p]$) and its covariance matrix C. This lower dimensional space contains linear combinations of original features called principal components (PCs; i.e., $Z = a^TX$, where $a^T= [a_1, a_2, \ldots, a_p]$, $Z^T= [Z_1, Z_2, \ldots, Z_p]$). The first PC, Z_1, is obtained by identifying a projection vector a_1 that maximizes the variance Z_1 equivalent to maximize $a_1^T C a_1$. The second PC, Z_2, is obtained to find a_2 that maximizes the variance Z_2 with a constraint that Z_1 and Z_2 are orthogonal. This process is repeated p times to obtain p PCs. Let E ($E^T= [E_1, E_2, \ldots, E_p]$) is a set of eigenvectors of C, with corresponding ordered eigenvalues ($\lambda_1 > \lambda_2 > \ldots > \lambda_p$). Using some properties of linear algebra, the projection vectors are E. That is, the PCs of X are calculated by the transformation process, $Z=EX$. PCA is efficient in facilitating visualization of high-dimensional spectral data if only the first few PCs are needed to represent most of the variability in the original high-dimensional space. However, PCA has several limitations in the analysis of spectral data. First, extracted PCs are linear combinations of a large number of original features. Thus, PCs cannot be readily interpreted. Second, the PCs may not produce maximum discrimination between sample classes because the transformation process of PCA relies solely on input variables and ignores class information. Third, determination of the number of PCs to retain is subjective.

The contribution of each feature to form PCs can be represented using loading vectors $\boldsymbol{k}$.

$$PC_i = \mathbf{x}_1 k_{i1} + \mathbf{x}_2 k_{i2} + \ldots + \mathbf{x}_N k_{ip} = \mathbf{X} k_i,\ i = 1, 2, \ldots, p. \quad (1)$$

For example, k_{i1} indicates the degree of importance of the first feature in the i^{th} PC domain. Typically, the first few PCs are sufficient to account for most of the variability in the data. Thus, a PCA loading value for the j^{th} original feature can be computed from the first k *PC*s.

$$Loading_j^{PCA} = \sum_{i=1}^{k} |k_{ij}| \omega_i,\ j=1, 2, \ldots, p, \quad (2)$$

where p is the total number of features of interest and ω_i represents the weight of the i^{th} PC. The simple way to determine ω_i is to compute the proportion of total variance explained by i^{th} PC.

Partial Least Squares: Similar to PCA, PLS uses transformed variables to identify a lower dimensional variable set (Boulesteix & Strimmer, 2007). PLS is a multivariate projection method for modeling a relation-

ship between independent variables **X** and dependent variable(s) **Y**. PLS seeks to find a set of latent features that maximizes the covariance between **X** and **Y**. It decomposes **X** and **Y** into the following forms:

$$\mathbf{X} = \mathbf{T}\mathbf{P}^T + \mathbf{E}, \quad (3)$$

$$\mathbf{Y} = \mathbf{T}\mathbf{Q}^T + \mathbf{F}, \quad (4)$$

where **T** is ($n \times A$) the score matrix of the extracted A score vectors that can be represented as $\mathbf{T}=\mathbf{X}\mathbf{w}$. **P** ($N \times A$) and **Q** ($M \times A$) loading matrices, and **E** ($n \times N$) and **F** ($n \times M$) residual matrices. The PLS method searches for weight vector **w** that maximizes the covariance between **Y** and **T.** By regressing **T** on **Y**, **Q** can be computed as follows:

$$\mathbf{Q} = (\mathbf{T}^T\mathbf{T})^{-1}\mathbf{T}^T\mathbf{Y}. \quad (5)$$

Then, the PLS regression model can be expressed as $\mathbf{Y} = \mathbf{X}\mathbf{B} + \mathbf{G}$, where $\mathbf{B} (=\mathbf{w}\mathbf{Q}^T)$ and **G** represent regression coefficients and a residual matrix, respectively. When the response variable is available in the spectral data, transformed variables obtained from PLS are more informative than those from PCA because the formal variables account for the distinction among different classes. However, the reduced dimensions from PLS do not provide a clear interpretation with respect to the original features for the same reason described for PCA

Fisher Discriminant Function: Fisher discriminant analysis (FDA) is a widely used technique for achieving optimal dimension reduction in classification problems in which the response variables are categorical. FDA provides an efficient lower dimensional representation of **X** for discrimination among groups. In other words, FDA uses the dependent variable to seek directions that are optimal for discrimination. This process is achieved by maximizing the between-group-scatter matrix $\mathbf{S}_b$ (see Equation (6)) while minimizing the within-group-scatter matrix $\mathbf{S}_w$ (see Equation (7)) (Yang et al., 2004). Thus, FDA finds optimal discriminant weight vectors φ by maximizing the following Fisher criterion:

$$J(\varphi) = \frac{\varphi^T \mathbf{S}_b \varphi}{\varphi^T \mathbf{S}_w \varphi}. \quad (6)$$

It can be shown that this maximization problem can be reduced to a generalized eigenvalue problem: $\mathbf{S}_b\boldsymbol{\varphi} = \lambda\mathbf{S}_w\boldsymbol{\varphi}$ (Yang et al., 2004). The main idea of using FDA for feature extraction is to use its weight vector (i.e., $\boldsymbol{\varphi} = [\phi_1, \phi_2, ..., \phi_N]^T$). FDA is the most efficient because transformed features provide better classification capability. However, by using FDA the feature extraction process may encounter computational difficulty because of the singularity of the scatter matrix when the number of samples is smaller than the number of features (Chen et al., 2000). Furthermore, the extracted features have the same interpretation problem as PCA and PLS.

The features selected by PCA, PLS, and FDA are evaluated and compared based on an ability of classification of high-resolution NMR spectra (Cho et al., in press).

Feature Selection Techniques

Genetic Algorithm: Interpretation problems posed by the transformation process in feature extraction can be overcome by selecting the best subset of given features in a spectrum. Genetic algorithms (GAs) have been successfully used as an efficient feature selection technique for PLS-based calibration (e.g., Esteban-Díez et al., 2006) and for other problems such as classification and feature selection for calibration (e.g., Kemsley, 2001). Recently, a two-stage genetic programming method was developed for selecting important features in NMR spectra for the classification of genetically modified barley (Davis et al., 2006). A major advantage of using GAs for feature selection in spectral data is that GAs take into account autocorrelation of features that make the selected features less dispersed. In other words, if the feature p is selected, features p-1 and p+1 are selected with high probability. However, GAs are known to be vulnerable to overfitting when the features are noisy or the dimension of the features is high. Also, too many parameters in GAs often prevent us from obtaining robust results.

Multiple Hypothesis Testing: The feature selection problem in NMR spectra can be formulated by constructing multiple hypothesis tests. Each null hypothesis states that the average intensities of the i^{th} features are equal among different experimental conditions, and the alternative hypothesis is that they differ. In multiple testing problems, it is traditional to choose a p-value threshold

τ and declare the feature x_j significant if and only if the corresponding *p*-value $p_j \leq \tau$. Thus, it is important to find an appropriate threshold that compromises the tradeoff between false positives and false negatives. A naive threshold is to use the traditional *p*-value cutoff of $\alpha = 0.01$ or 0.05 for individual hypothesis. However, applying a single testing procedure to a multiple testing problem leads to an exponential increase in false positives. To overcome this, the Bonferroni method uses a more stringent threshold (Shaffer, 1995). However, the Bonferroni method is too conservative and often fails to detect the truly significant feature. In a seminal paper, Benjamini & Hochberg (1995) proposed an efficient procedure to determine a suitable threshold that controls the false discovery rate (FDR), defined as the expected proportion of false positives among all the hypotheses rejected. A summary of the procedure is as follows:

- Select a desired FDR level (=α) between 0 and 1.
- Find the largest *i* denoted as w, where

$$w = \max\left[i : p(i) \leq \frac{i}{m}\frac{\alpha}{\delta}\right].$$

  m is the total number of hypotheses, and δ denotes the proportion of true null hypotheses. Several studies discuss the assignment of δ where δ=1 is the most conservative choice.
- Let the *p*-value threshold be $p_{(w)}$, and declare the metabolite feature t_i significant if and only if $p_i \leq p_{(w)}$.

The FDR-based procedure treats all the features simultaneously by constructing multiple hypotheses and the ability of the procedure to control FDR provides the logical interpretation of feature selection results. In addition, the effect of correlated features in a spectrum can be automatically accommodated by the FDR procedure through a simple modification (Benjamini & Yekutieli, 2001).

To demonstrate the effectiveness of the FDR procedure, we conduct a simple case study using 136 NMR spectra (each with 574 metabolite features) extracted from two different experimental conditions. A multiple testing procedure controlling FDR (at α= 0.01) is performed to find the potentially significant metabolite features that differentiate two different experimental conditions. The FDR procedure identifies 75 metabolite features as potentially significant ones. To demonstrate the advantage of FDR, we compare the classification capability between the features selected by FDR and full features using a *k*-nearest neighbor (KNN) algorithm. Different values of *k* are run with Euclidean distance to find the optimal KNN model that has minimum misclassification rate. The results show the KNN model constructed with the features selected by FDR yield smaller misclassification rate (32%) than the model with full features (38%). This implies that feature selection by FDR adequately eliminates non-informative features for classification.

FUTURE TRENDS

Developing highly efficient methods to identify meaningful features from spectral data is a challenge because of their complicated formation. The processes of the presented features extraction/selection methods are performed on original spectra that are considered a single scale, although the optimal scales of the spectral regions associated with different chemical species vary. In other words, spectral data usually comprise a number of local bumps or peaks with different scales (i.e., peaks of different sizes). Thus, efficient analytical tools are needed to handle the multiscale nature of NMR spectra. Wavelet transforms are the most useful method to use with NMR spectra because such transforms have the advantage of locality of analysis and the capability to handle multiscale information efficiently (Bruce et al., 2002). Wavelet transforms have numerous advantages over conventional feature extraction methods. First, the extracted coefficients from wavelet transforms are combinations of a small number of local features that can provide the physical meaning of features in spectra. Second, the capability of wavelets to break down the original spectra into different resolution levels can efficiently handle the multiscale nature of spectral data. Third, wavelets have the highly desirable property of reducing the serial correlation between the features in a spectrum (Wornell, 1996). Finally, wavelet transforms coupled with a relevant thresholding method can remove noise, thus, enhancing the feature selection process.

CONCLUSION

This chapter presents feature extraction and feature selection methods for identifying important features in spectral data. The present feature extraction and feature selection methods have their own advantages and disadvantages, and the choice between them depends upon the purpose of the application. The transformation process in feature extraction takes fully into account the multivariate nature of spectral data and thus provides better discriminative ability but suffers from the lack of interpretability. This problem of interpretation may be overcome with recourse to feature selection because transformations are absent from the process, but then the problem arises of incorporating the correlation structure present in spectral data. Wavelet transforms are a promising approach as the compromise between feature extraction and feature selection. Extracted coefficients from wavelet transforms may have a meaningful physical interpretation because of their locality. Thus, minimal loss of interpretation occurs when wavelet coefficients are used for feature selection.

REFERENCES

Beckonert, O., Bollard, M. E., Ebbels, T. M. D., Keun, H. C., Antti, H., Holmes, E., et al. (2003). NMR-based metabonomic toxicity classification: Hierarchical cluster analysis and k-nearest-neighbour approaches. *Analytica Chimica Acta, 490*(1-2), 3-15.

Benjamini, Y., & Hochberg, Y. (1995). Controlling the false discovery rate - a practical and powerful approach to multiple testing. *Journal Of The Royal Statistical Society Series B-Methodological, 57*(1), 289-300.

Benjamini, Y., & Yekutieli, D. (2001). The control of the false discovery rate in multiple testing under dependency. *Annals Of Statistics, 29*(4), 1165-1188.

Boulesteix, A.-L., & Strimmer, K. (2007). Partial least squares: A versatile tool for the analysis of high-dimensional genomic data. *Briefings in Bioinformatics, 8*, 32-44.

Bruce, L. M., Koger, C. H., & Li, J. (2002). Dimensionality reduction of hyperspectral data using discrete wavelet transform feature extraction. *IEEE Geoscience and Remote Sensing, 40*, 2331-2338.

Chen, L.-F., Lio, H.-Y. M., Ko, M.-T., Lin, J.-C., & Yu, G.-J. (2000). A new lda-based face recognition system which can solve the small sample size problem. *Pattern Recognition, 33*, 1713-1726.

Cho, H.-W., Kim, S. B. K., Jeong, M., Park, Y., Gletsu, N., Ziegler, T. R., et al. (in press). Discovery of metabolite features for the modeling and analysis of high-resolution NMR spectra. *International Journal of Data Mining and Bioinformatics*.

Davis, R. A., Charlton, A. J., Oehlschlager, S., & Wilson, J. C. (2006). Novel feature selection method for genetic programming using metabolomic ^{1}H NMR data. *Chemometrics And Intelligent Laboratory Systems, 81*, 50-59.

Dunn, W. B., & Ellis, D. I. (2005). Metabolomics: Current analytical platforms and methodologies. *Trends in Analytical Chemistry, 24*, 285-294.

Esteban-Díez, I., González-Sáiz, J. M., Gómez-Cámara, D., & Pizarro Millan, C. (2006). Multivariate calibration of near infrared spectra by orthogonal wavelet correction using a genetic algorithm. *Analytica Chimica Acta, 555*, 84-95.

Holmes, E., & Antti, H. (2002). Chemometric contributions to the evolution of metabonomics: Mathematical solutions to characterising and interpreting complex biological nmr spectra. *Analyst, 127*(12), 1549-1557.

Holmes, E., Nicholson, J. K., & Tranter, G. (2001). Metabonomic characterization of genetic variations in toxicological and metabolic responses using probabilistic neural networks. *Chemical Research In Toxicology, 14*(2), 182-191.

Jiménez-Rodríguez, L. O., Arzuaga-Cruz, E., & Vélez-Reyes, M. (2007). Unsupervised linear feature-extraction methods and their effects in the classification of high-dimensional data. *IEEE Geoscience and Remote Sensing, 45*, 469-483.

Jollife, I. T. (2002). *Principal component analysis (second edition)*. New York, NY: Springer.

Kemsley, E. K. (2001). A hybrid classification method: Discrete canonical variate analysis using a genetic algorithm. *Chemometrics and Intelligent Laboratory Systems, 55*, 39-51.

Lindon, J. C., Holmes, E., & Nicholson, J. K. (2001). Pattern recognition methods and applications in bio-

medical magnetic resonance. *Progress In Nuclear Magnetic Resonance Spectroscopy, 39*(1), 1-40.

Nicholson, J. K., Connelly, J., Lindon, J. C., & Holmes, E. (2002). Metabonomics: A platform for studying drug toxicity and gene function. *Nature Reviews Drug Discovery, 1*(2), 153-161.

Shaffer, J. P. (1995). Multiple hypothesis-testing. *Annual Review Of Psychology, 46*, 561-584.

Wornell, G. W. (1996). Emerging applications of multirate signal processing and wavelets in digintal communications. *Proceedings Of IEEE, 84*, 586-603.

Yang, J., Frangia, A. F., & Yang, J. (2004). A new kernel fisher discriminant algorithm with application to face recognition. *Neurocomputing, 56*, 415-421.

KEY TERMS

False Discovery Rate: The expected proportion of false positives among all the hypotheses rejected.

Feature: Measurable values that characterize an individual observation.

Feature Extraction: The process of creating new features based on transformations of the original features.

Feature Selection: The process of identifying a subset of important features that leads to the best prediction or classification.

Metabolomics: The global analysis for the detection and recognition of metabolic changes in response to pathophysiological stimuli, genetic modification, nutrition intake, and toxins in integrated biological systems.

Nuclear Magnetic Resonance Spectroscopy: The technique that uses the magnetic properties of nuclei to understand the structure and function of chemical species.

Wavelet Transforms: The process of approximating a signal at different scales (resolutions) and space simultaneously.

Feature Reduction for Support Vector Machines

Shouxian Cheng
Planet Associates, Inc., USA

Frank Y. Shih
New Jersey Institute of Technology, USA

INTRODUCTION

The *Support Vector Machine* (SVM) (Cortes and Vapnik,1995; Vapnik, 1995; Burges, 1998) is intended to generate an optimal separating hyperplane by minimizing the generalization error without the assumption of class probabilities such as Bayesian classifier. The decision hyperplane of SVM is determined by the most informative data instances, called *Support Vectors* (SVs). In practice, these SVMs are a subset of the entire training data. By now, SVMs have been successfully applied in many applications, such as face detection, handwritten digit recognition, text classification, and data mining. Osuna et al. (1997) applied SVMs for face detection. Heisele *et al*. (2004) achieved high face detection rate by using 2nd degree SVM. They applied hierarchical classification and feature reduction methods to speed up face detection using SVMs.

Feature extraction and reduction are two primary issues in feature selection that is essential in pattern classification. Whether it is for storage, searching, or classification, the way the data are represented can significantly influence performances. Feature extraction is a process of extracting more effective representation of objects from raw data to achieve high classification rates. For image data, many kinds of features have been used, such as raw pixel values, Principle Component Analysis (PCA), Independent Component Analysis (ICA), wavelet features, Gabor features, and gradient values. Feature reduction is a process of selecting a subset of features with preservation or improvement of classification rates. In general, it intends to speed up the classification process by keeping the most important class-relevant features.

BACKGROUND

Principal Components Analysis (PCA) is a multivariate procedure which rotates the data such that the maximum variabilities are projected onto the axes. Essentially, a set of correlated variables are transformed into a set of uncorrelated variables which are ordered by reducing the variability. The uncorrelated variables are linear combinations of the original variables, and the last of these variables can be removed with a minimum loss of real data. PCA has been widely used in image representation for dimensionality reduction. To obtain m principal components, a transformation matrix of $m \times N$ is multiplied by an input pattern of $N \times 1$. The computation is costly for high dimensional data.

Another well-known method of feature reduction uses Fisher's criterion to choose a subset of features that possess a large between-class variance and a small within-class variance. For two-class classification problem, the within-class variance for i-th dimension is defined as

$$\sigma_i^2 = \frac{\sum_{j=1}^{l} (g_{j,i} - m_i)^2}{l-1}, \tag{1}$$

where l is the total number of samples, $g_{j,i}$ is the i-th dimensional attribute value of sample j, and m_i is the mean value of the i-th dimension for all samples. The Fisher's score for between-class measurement can be calculated as

$$S_i = \left| \frac{m_{i,class1} - m_{i,class2}}{\sigma_{i,class1}^2 + \sigma_{i,class2}^2} \right| . \tag{2}$$

By selecting the features with the highest Fisher's scores, the most discriminative features between class 1 and class 2 are retained.

Weston et al. (2000) developed a feature reduction method for SVMs by minimizing the bounds on the leave-one-out error. Evgenious et al. (2003) introduced a method for feature reduction for SVMs based on the observation that the most important features are the ones that separate the hyperplane the most. Shih and Cheng (2005) proposed an improved feature reduction method in input and feature space for the 2nd degree polynomial SVMs.

MAIN FOCUS

In this section, we present an improved feature reduction method for the 2nd degree polynomial SVMs. In the input space, a subset of input features is selected by ranking their contributions to the decision function. In the feature space, features are ranked according to the weighted support vector in each dimension. By applying feature reduction in both input and feature space, a fast non-linear SVM is designed without a significant loss in performance. Here, the face detection experiment is used to illustrate this method.

Introduction to Support Vector Machines

Consider a set of l labeled training patterns $(\mathbf{x}_1, y_1)$, $(\mathbf{x}_2, y_2)$, ..., $(\mathbf{x}_i, y_i)$, ..., $(\mathbf{x}_l, y_l)$ where $\mathbf{x}_i$ denotes the i-th training sample and $y_i \in \{1, -1\}$ denotes the class label. For two-class classification, SVMs use a hyperplane that maximizes the margin (i.e., the distance between the hyperplane and the nearest sample of each class). This hyperplane is viewed as the *Optimal Separating Hyperplane* (OSH).

If the data are not linearly separable in the input space, a non-linear transformation function $\Phi(\cdot)$ is used to project $\mathbf{x}_i$ from the input space to a higher dimensional feature space. An OSH is constructed in the feature space by maximizing the margin between the closest points $\Phi(\mathbf{x}_i)$ of two classes. The inner-product between two projections is defined by a kernel function $K(\mathbf{x}, \mathbf{y}) = \Phi(\mathbf{x}) \cdot \Phi(\mathbf{y})$. The commonly-used kernels include polynomial, Gaussian RBF, and Sigmoid kernels.

The decision function of the SVM is defined as

$$f(\mathbf{x}) = \mathbf{w} \cdot \Phi(\mathbf{x}) + b = \sum_{i=1}^{l} \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b\,, \tag{3}$$

where $\mathbf{w}$ is the support vector, α_i is the Lagrange multiplifier, and b is a constant. The optimal hyperplane can be obtained by maximizing

$$W(\boldsymbol{\alpha}) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i\, \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)\,, \tag{4}$$

subject to

$$\sum_{i=1}^{l} \alpha_i y_i = 0 \text{ and } 0 \le \alpha_i \le C,$$

where C is regularization constant that manages the tradeoff between the minimization of the number of errors and the necessity to control the capacity of the classifier.

Feature Reduction in Input Space

A feature reduction method is proposed for the 2nd-degree polynomial SVM with kernel $K(\mathbf{x}, \mathbf{y}) = (1 + \mathbf{x} \cdot \mathbf{y})^2$ After training, the decision function for a pattern $\mathbf{x}$ is defined in Box 1, where s is the total number of support vectors, $\mathbf{x}_i$ is the i-th support vector, and $x_{i,k}$ and x_k are respectively the k-th dimension for the support vector $\mathbf{x}_i$ and the pattern $\mathbf{x}$. The component in the k-th dimension (where $k = 1, 2, \ldots, N$) is shown in Box 2.

The m features with the largest contributions to the decision function are selected from the original N features. The contribution can be obtained by

$$F(k) = \int_V f(\mathbf{x}, k)\, dP(\mathbf{x}), \tag{7}$$

where V denotes the input space and $P(\mathbf{x})$ denotes the probability distribution function. Since $P(\mathbf{x})$ is unknown, we approximate $F(k)$ using a summation over the support vectors as shown in Box 3.

Box 1.

$$f(\mathbf{x}) = \sum_{i=1}^{s} \alpha_i y_i (1 + \mathbf{x}_i \cdot \mathbf{x})^2 + b$$

$$(= \sum_{i=1}^{s} \alpha_i y_i \ 1 + x_{i,1}x_1 + x_{i,2}x_2 + \ldots + x_{i,k}x_k + \ldots + x_{i,N}x_N)^2 + b \quad (5)$$

Box 2.

$$f(\mathbf{x}, k) = \sum_{i=1}^{s} \alpha_i y_i \left[2x_k x_{i,k} (1 + x_{i,1}x_1 + \ldots + x_{i,k-1}x_{k-1} + x_{i,k+1}x_{k+1} + \ldots + x_{i,N}x_N) + x_k^2 x_{i,k}^2 \right]$$

$$2 = \sum_{i=1}^{s} \alpha_i y_i \left[\ x_k x_{i,k} (1 + x_{i,1}x_1 + \ldots + x_{i,N}x_N) - x_k^2 x_{i,k}^2 \right]. \quad (6)$$

Box 3.

$$F(k) = \sum_{i=1}^{s} \left| \sum_{j=1}^{s} \alpha_j y_j [2x_{i,k}x_{j,k}(1 + x_{j,1}x_{i,1} + \ldots + x_{j,N}x_{i,N}) - x_{i,k}^2 x_{j,k}^2] \right| \quad (8)$$

For experiment, a face image database from the Center for Biological and Computational Learning at Massachusetts Institute of Technology (MIT) is adopted, which contains 2,429 face training samples, 472 face testing samples, and 23,573 non-face testing samples. The 15,228 non-face training samples are randomly collected from the images that do not contain faces. The size of all these samples is 19 × 19.

In order to remove background pixels, a mask is applied to extract only the face. Prior to classification, we perform image normalization and histogram equalization. The image normalization is used to normalize the gray-level distribution by the Gaussian function with zero mean and one variance. Histogram equalization uses a transformation function equal to the cumulative distribution to produce an image whose gray levels have a uniform density. Figure 1 shows a face image, the mask, the image after normalization, and the image after histogram equalization.

Different preprocessing methods have been tested: image normalization, histogram equalization, and without preprocessing. We obtain that using normalization or equalization can produce almost the same but much better results than without preprocessing. Therefore, in the following experiments, image normalization is used as the pre-processing method. To calculate PCA values, only positive training samples are used to calculate transformation matrix. The reason is that only training face samples are used in the calculation of the transformation matrix, and for testing the input samples are projected on the face space. Therefore, better classification results can be achieved in separating face and non-face classes

Using the normalized 2,429 face and 15,228 non-face training samples and taking all the 283 gray values as input to train the 2nd-degree SVM, we obtain 252 and 514 support vectors for face and non-face classes, respectively. Using these support vectors in Eq. (8), we obtain $F(k)$, where k = 1, 2, …, 283. Figure 2 shows the Receiver Operating Characteristic (ROC) curves for different kinds of features in input space. The ROC curve is defined as shifting the SVM hyperplane by changing the threshold value b to obtain corresponding detection rates and false positive rates. Face classification is performed on the testing set to calculate the false positive and the detection rates. The horizontal

Figure 1.

(a) Original face image

(b) the mask

(c) normalized image

(d) histogram equalized image

axis shows the false positive rate over 23,573 non-face testing samples. The vertical axis shows the detection rate over 472 face testing samples.

Using 100 gray values selected by the ranking method is compared with the methods of using all the 283 gray values, 100 PCA features, Fisher's scores, and the 100 features selected by Evgenious *et al.* (2003). It is observed that our method performs similarly as using all the 283 features and using 100 PCA values; however, it is better than using the 100 features by Fisher's scores and by Evgenious *et al.* (2003).

Feature Reduction in Feature Space

In feature space, the decision function $f(\mathbf{x})$ of SVMs is defined as

$$f(\mathbf{x}) = \sum_{i=1}^{s} \alpha_i y_i (\Phi(\mathbf{x}) \cdot \Phi(\mathbf{x}_i)) + b = \mathbf{w} \cdot \Phi(\mathbf{x}) + b, \tag{9}$$

where $\mathbf{w}$ is the support vector. For a 2nd-degree polynomial SVM with the input space of dimension N and kernel $K(\mathbf{x}, \mathbf{y}) = (1 + \mathbf{x} \cdot \mathbf{y})^2$ the feature space is given by

$$\Phi(\mathbf{x}) = (\sqrt{2}x_1, ..., \sqrt{2}x_N, x_1^2, ..., x_N^2, \sqrt{2}x_1x_2, ..., \sqrt{2}x_{N-1}x_N) \tag{10}$$

of dimension $P = N(N+3)/2$.

Suppose a 2nd-degree SVM using face and non-face samples is trained to obtain s support vectors. The support vector in the feature space can be represented as

$$\mathbf{w} = \sum_{i=1}^{s} \alpha_i y_i \Phi(\mathbf{x}_i) = (w_1, w_2, ..., w_P). \tag{11}$$

One way to select a subset of features is to rank $|w_k|$, for $k = 1, 2, ..., P$. An improved method of using is proposed,

$$\left| w_k \int_V | x_k^* | \, dp(x_k^*) \right|$$

where x_k^* denotes the k-th dimension of $\mathbf{x}$ in the feature space V. Since the distribution function $dp(x_k^*)$ is unknown, the ranking function $R(k)$ is calculated as

$$R(k) = \left| w_k \sum_{i=1}^{s} | x_{i,k}^* | \right|, \tag{12}$$

where $x_{i,k}^*$ denotes the k-th dimension of $\mathbf{x}_i$. The decision function of q features is calculated as

$$f(\mathbf{x}, q) = \mathbf{w}(q) \cdot \Phi(\mathbf{x}, q) + b,$$

where $\mathbf{w}(q)$ is the selected q features in $\mathbf{w}$ and $\Phi(\mathbf{x}, q)$ is the corresponding q features in $\mathbf{x}$.

In the experiments, a 2nd-degree polynomial SVM is trained using 60 PCA values in the input space. We obtain 289 and 412 support vectors for face and non-face classes, respectively. The 1,890 features in the feature space can be calculated by Eq. (10). The support vector in the feature space, $\mathbf{w} = (w_1, w_2, ..., w_{1890})$ can be calculated by Eq. (11). Given a pattern $\mathbf{x}$, the decision value of using the selected q features can be calculated by Eq. (13). When q = 300, 500, and 1000, we illustrate the results in Figure 3. It is observed that using the selected 500 or 1,000 features can achieve almost the same performance as using all the 1,890 features. However, using 300 features is insufficient to achieve a good performance.

Figure 2. ROC curves for comparisons of using different features in input space

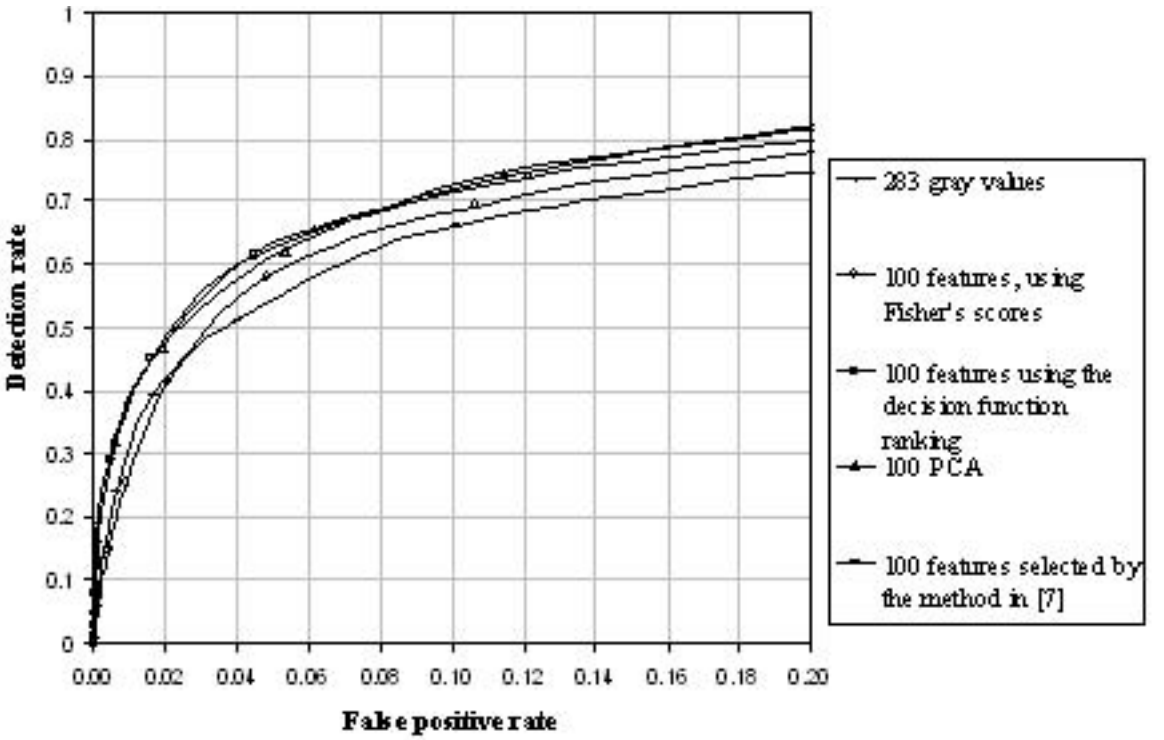
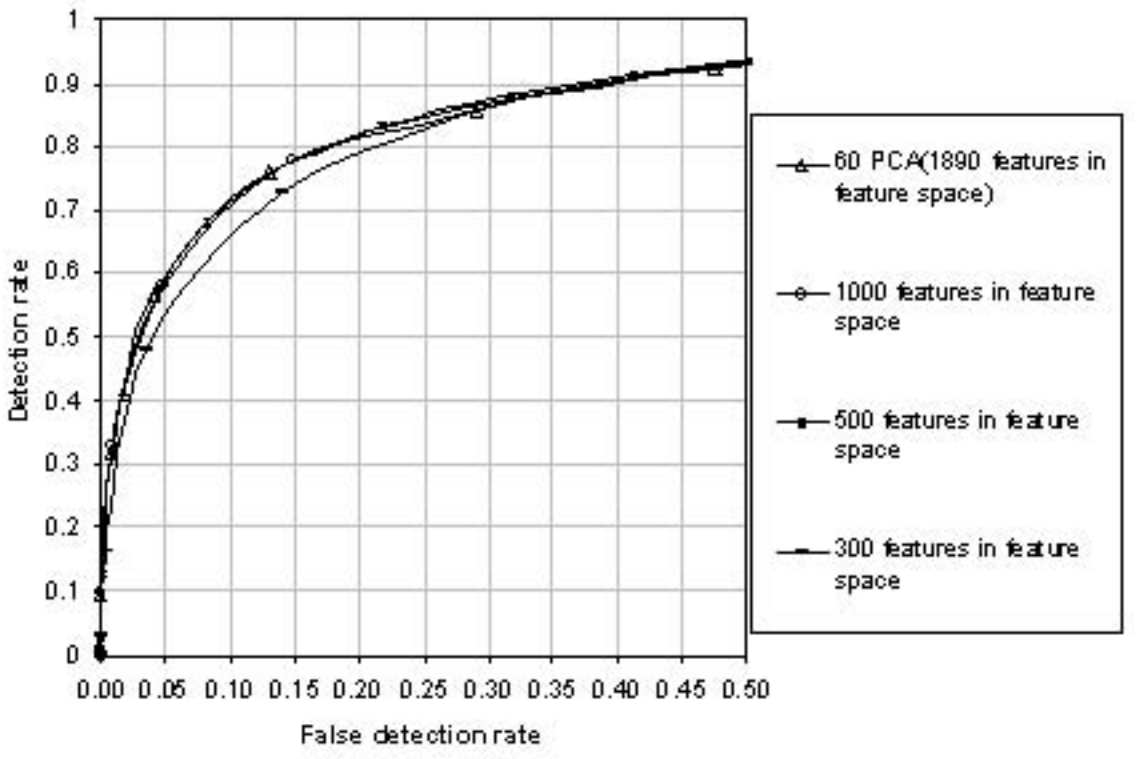


Figure 3. ROC curves for different numbers of features in the feature space



Combinational Feature Reduction in Input and Feature Spaces

First, we choose m features from the N input space, and train the 2nd-degree polynomial SVM. Next, we select q features from $P = m(m + 3)/2$ features in the feature space to calculate the decision value. We use two following methods to compute the decision values: one by Eq. (3) in input space and the other by Eq. (5) in feature space. In Eq. (3), the number of multiplications required to calculate the decision function is $(N + 1)s$. Note that s is the total number of support vectors. In Eqs. (5) , the total number of multiplications required is $(N + 3)N$. If $(N + 1)s > (N + 3)N$, it is more efficient to implement the 2nd-degree polynomial SVM in the feature space; otherwise, in the input space. This is evidenced by our experiments because the number of support vectors is more than 700, that is much larger than N. Note that $N = 283$, 60, or 100 indicates all the gray-value features, 60 PCA values, or 100 features, respectively.

The SVM is trained using the selected 100 features as described above and 244 and 469 support vectors are obtained for face and non-face classes, respectively. Fig. 4 shows the comparisons of our combinational method using 3,500 features in feature space, 60 PCA values, and all the 283 gray values. It is observed that using our combinational method can obtain competitive results as using 60 PCA values or all 283 gray values. Apparently, our method gains the advantage of speed.

Table 1 lists the number of features used in the input and feature space and the number of multiplications required in calculating the decision values for comparing our method with the methods of using all the gray values and using PCA features.

From Table 1, it is observed that using PCA for feature reduction in the input and feature space, a speed-up factor of 4.08 can be achieved. However, using our method, a speed-up factor of 9.36 can be achieved. Note that once the features in the feature space are determined, we do not need to project the input space on the whole feature space, but on the selected feature space. This

Figure 4. ROC curves of using the proposed feature reduction method, 60 PCA, and all the 283 gray values

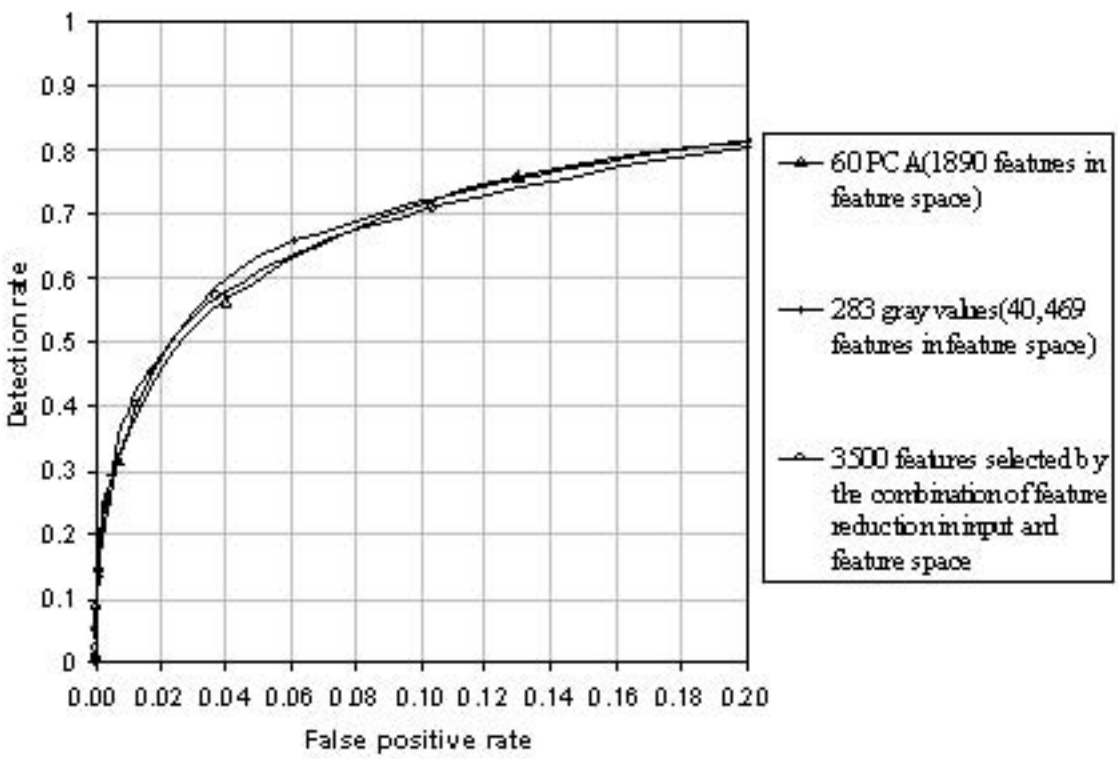


Table 1. Comparisons of the number of features and the number of multiplications

Methods	Number of features in input space	Number of features in feature space	Number of multiplications
All gray values	283	40,469	80,938
PCA	60	1,890	20,760
Our method	100	3,500	8,650

can further reduce the computation, i.e., only 7,000 multiplications are required instead of 8,650.

FUTURE TRENDS

SVMs have been widely used in many applications. In many cases, SVMs have achieved either better or similar performance as other competing methods. Due to the time-consuming training procedure and high dimensionality of data representation, feature extraction and feature reduction will continue to be important research topics in pattern classification field. An improved feature reduction method has been proposed by combining both input and feature space reduction for the 2nd degree SVM. Feature reduction methods for Gaussian RBF kernel need to be studied. Also it is an interesting research topic to explore how to select and combine different kind features to improve classification rates for SVMs.

CONCLUSION

An improved feature reduction method in the combinational input and feature space for Support Vector Machines (SVMs) has been described. In the input space, a subset of input features is selected by ranking their contributions to the decision function. In the feature space, features are ranked according to the weighted support vector in each dimension. By combining both input and feature space, we develop a fast non-linear SVM without a significant loss in performance. The proposed method has been tested on the detection of face, person, and car. Subsets of features are chosen from pixel values for face detection and from Haar wavelet features for person and car detection. The experimental results show that the proposed feature reduction method works successfully.

REFERENCES

Burges, C. J.C. (1998). A Tutorial on support vector machines for pattern recognition, *IEEE Trans. Data Mining and Knowledge Discovery*, 2(2), 121-167.

Cortes, C., and Vapnik, V. (1995). Support-vector networks, *Machine Learning*, 20(3), 273-297.

Evgenious, T., Pontil, M., Papageorgiou, C., and Poggio, T. (2003). Image representations and feature selection for multimedia database search, *IEEE Trans. Knowledge and Data Engineering*, 15(4), 911-920.

Heisele, B., Serre, T., Prentice, S., and Poggio, T. (2003). Hierarchical classification and feature reduction for fast face detection with support vector machines, *Pattern Recognition*, 36(9), 2007-2017.

Osuna, E., Freund, R., and Girosi, F. (1997). Training support vector machines: an application to face detection, *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 130-136.

Shih, F. Y., and Cheng, S. (2005). Improved feature reduction in input and feature spaces, *Pattern Recognition*, 38(5), 651-659.

Vapnik, V. (1998). *Statistical Learning Theory*, Wiley, New York.

Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., and Vapnik, V. (2000). Feature selection for support vector machines, *Advances in Neural Information Processing System*, 13, 668-674.

KEY TERMS

Feature Extraction: Feature extraction is a process of extracting more effective representation of objects from raw data to achieve high classification rates.

Feature Reduction: Feature reduction is a process of selecting a subset of features with preservation or improvement of classification rates.

Machine Learning: As a subfield of artificial intelligence, machine learning is concerned with the development of algorithms and techniques that allow computers to learn and to recognize patterns that have occurred repeatedly and improve its performance based on past experience.

Principle Component Analysis (PCA): PCA is a multivariate procedure which rotates the data such that the maximum variabilities are projected onto the axes. PCA has been widely used in image representation for dimensionality reduction.

Receiver Operating Characteristic (ROC) curves: The ROC curve for SVM is defined as shifting the SVM hyperplane by changing the threshold value b to obtain corresponding detection rates and false positive rates.

Support Vector Machine: For two-class classification, SVMs use a hyperplane that maximizes the margin (i.e., the distance between the hyperplane and the nearest sample of each class). This hyperplane is viewed as the *Optimal Separating Hyperplane* (OSH).

Feature Selection

Damien François
Université catholique de Louvain, Belgium

INTRODUCTION

In many applications, like function approximation, pattern recognition, time series prediction, and data mining, one has to build a model relating some features describing the data to some response value. Often, the features that are relevant for building the model are not known in advance. Feature selection methods allow removing irrelevant and/or redundant features to only keep the feature subset that are most useful to build a prediction model. The model is simpler and easier to interpret, reducing the risks of overfitting, non-convergence, etc. By contrast with other dimensionality reduction techniques such as principal component analysis or more recent nonlinear projection techniques (Lee & Verleysen 2007), which build a new, smaller set of features, the features that are selected by feature selection methods preserve their initial meaning, potentially bringing extra information about the process being modeled (Guyon 2006).

Recently, the advent of high-dimensional data has raised new challenges for feature selection methods, both from the algorithmic point of view and the conceptual point of view (Liu & Motoda 2007). The problem of feature selection is exponential in nature, and many approximate algorithms are cubic with respect to the initial number of features, which may be intractable when the dimensionality of the data is large. Furthermore, high-dimensional data are often highly redundant, and two distinct subsets of features may have very similar predictive power, which can make it difficult to identify the best subset.

BACKGROUND

Feature selection methods are often categorized as 'filters,' 'wrappers,' or 'embedded' methods. Roughly stated, filters use statistical measures to 'filter out' unneeded features before building the model. Wrappers, by contrast, use the prediction error of the model to select the features. Embedded methods are actually prediction models that propose, as a by-product, a scoring, or even a selection of the features; like for instance decision trees (Breiman, 2001) and LASSO models (Efron, 2004).

This distinction between filters and wrappers, which has historical roots (Kohavi & John, 1997), is less and less relevant now because many new methods are very difficult to label with either name. More generally, all feature selection methods share the same structure; they need (1) a criterion that scores a feature or a set of features according to its (their) predictive power, and (2) a method to find the optimal subset of features according to the chosen criterion. This method comprises an exploration algorithm, which generates new subsets for evaluation, and a stopping criterion to help deciding when to stop the search.

Criteria for scoring a single feature include the well-known correlation, chi-squared measure, and many other statistical criteria. More powerful methods, like mutual information (Battiti, 1994), and the Gamma test (Stefansson *et al*, 1997) (for regression) and the RELIEF algorithm (Kira & Rendell, 1992) (for classification), allow scoring a whole subset of features. In a more wrapper-like approach, the performances of a prediction model can also be used to assess the relevance of a subset of features.

Algorithms for finding the optimal subset, which is a combinatorial problem, can be found in the Artificial Intelligence literature (Russel & Norvig, 2003). In the context of feature selection, greedy algorithms, which select or exclude one feature at a time, are very popular. Their reduced complexity still allows them to find near optimal subsets that are very satisfactory (Aha & Bankert, 1996).

MAIN FOCUS

This section discusses the concepts of relevance and redundancy, which are central to any feature selection method, and propose a step-by-step methodology along with some recommendations for feature selection on real, high-dimensional, and noisy data.

Relevance and Redundancy

What do we mean precisely by "relevant"? See Kohavi & John (1997); Guyon & Elisseeff (2003); Dash & Liu (1997); Blum & Langley (1997) for a total of nearly ten different definitions for the relevance of a feature subset. The definitions vary depending on whether a particular feature is relevant but not unique, etc. Counter-intuitively, a feature can be useful for prediction and at the same time irrelevant for the application. For example, consider a bias term. Conversely, a feature that is relevant for the application can be useless for prediction if the actual prediction model is not able to exploit it.

Is redundancy always evil for prediction? Surprisingly, the answer is no. First, redundant features can be averaged to filter out noise to a certain extent. Second, two correlated features may carry information about the variable to predict, precisely in their difference.

Can a feature be non relevant individually and still useful in conjunction with others? Yes. The typical example is the XOR problem (Y = X1 XOR X2), or the sine function over a large interval (Y = sin (2 π (X1 + X2)). Both features X1 and X2 are needed to predict Y, but each of them is useless alone; knowing X1 perfectly, for instance, does not allow deriving any piece of information about Y. In such case, only multivariate criteria (like mutual information) and exhaustive or randomized search procedures (like genetic algorithms) will provide relevant results.

A Proposed Methodology

Mutual information and Gamma test for feature subset scoring. The mutual information and the Gamma test are two powerful methods for evaluating the predictive power of a subset of features. They can be used even with high-dimensional data, and are theoretically able to detect any nonlinear relationship. The mutual information estimates the loss of entropy of the variable to predict, in the information-theoretical sense, when the features are known, and actually estimates the degree of independence between the variables. The mutual information, associated with a non-parametric test such as the permutation test, makes a powerful tool for excluding irrelevant features (François et al., 2007). The Gamma test produces an estimate of the variance of the noise that a nonlinear prediction model could reach using the given features. It is very efficient when totally irrelevant features have been discarded. Efficient implementations for both methods are available free for download (Kraskov et al, 2004; Stefansson et al, 1997).

Greedy subset space exploration. From the simple ranking (select the K features that have the most individual score), to more complex approaches like genetic algorithms (Yang & Honavar, 1998) or simulated annealing (Brooks et al, 2003), the potentially useful exploration techniques are numerous. Simple ranking is very efficient from a computational point of view, it is however not able to detect that two features are useful together while useless alone. Genetic algorithms, or simulated annealing, are able to find such features, at the cost of a very large computational burden.

Greedy algorithms are often a very suitable option. Greedy algorithms, such as Forward Feature Selection, work incrementally, adding (or removing) one feature at a time, and never questioning the choice of that feature afterwards. These algorithms, although being sub-optimal, often finds feature subsets that are very satisfactory, with acceptable computation times.

A step-by step methodology. Although there exists no ideal procedure that would work in all situations, the following practical recommendations can be formulated.

1. Exploit any domain knowledge to eliminate obviously useless features. Use an expert, either before selecting features to avoid processing obviously useless features. Removing two or three features a priori can make a huge difference for an exponential algorithm, but also for a cubic algorithm! You can also ask the expert after the selection process, to check whether the selected features make sense.
2. Perform simple ranking with the correlation coefficient. It is important to know if there is a strong linear link between features and the response value, because nonlinear models are seldom good at modeling linear mappings, and will certainly not outperform a linear model in such a case.

3. Perform simple ranking first using mutual information to get baseline results. It will identify the most individually relevant features, and maybe also identify features that are completely independent from the response value. Excluding them might be considered only if it leads to a significant reduction in computation time. Building a prediction model with the first three up to, say, the first ten relevant features, will give approximate baseline prediction results and will provide a rough idea of the computation time needed. This is an important piece of information that can help decide which options to consider.
4. If tractable, consider the exhaustive search of all pairs or triplets of features with a Gamma test and/or mutual information. This way, risks of ignoring important features that are useful together only, are reduced. If it is not tractable, consider only pairs or triplets involving at least one individually relevant feature, or even only relevant features. The idea is always to trade comprehensiveness of the search against computational burden.
5. Initiate a forward strategy with the best subset from previous step. Starting from three features instead of one will make the procedure less sensitive to the initialization conditions.
6. End with a backward search using the actual prediction model, to remove any redundant features and further reduce the dimensionality. It is wise at this stage to use the prediction model because (i) probably few iterations will be performed and (ii) this focuses the selection procedure towards the capabilities/flaws of the prediction model.
7. If the forward-backward procedure gives results that are not much better than those obtained with the ranking approach, resort to a randomized algorithm like the genetic algorithm first with mutual information or gamma test, then with the actual model if necessary. Those algorithms will take much more time than the incremental procedure but they might lead to better solutions, as they are less restrictive than the greedy algorithms.

Finally, the somewhat philosophical question "*What to do with features that are discarded ?"* can be considered. Caruana has suggested using them as extra outputs to increase the performances of the prediction model (Caruana & de Sa, 2003). These extra outputs force the model to learn the mapping from the selected features to the discarded ones. Another interesting idea is to use the discarded features, especially those discarded because of redundancy, along with some other relevant features to build one or several other models and group them into an ensemble model (Gunter & Bunke, 2002). This ensemble model combines the results of several different models (built on different feature subsets) to provide a more robust prediction.

FUTURE TRENDS

Feature selection methods will face data of higher and higher dimensionality, and of higher complexity, and therefore will need to adapt. Hopefully the process of feature selection can be relatively easily adapted for parallel architectures. Feature selection methods will also be tuned to handle structured data (graphs, functions, etc.). For instance, when coping with functions, intervals rather than single features should be selected (Krier, forthcoming).

Another grand challenge about feature selection is to assess the stability of the selected subset, especially if the initial features are highly correlated. How likely is the optimal subset to change if the data slightly change? Cross-validation and other resampling techniques could be used to that end, but at a computational cost that is, at the moment, not affordable.

CONCLUSION

The ultimate objective of feature selection is to gain both time and knowledge about a process being modeled for prediction. Feature selection methods aim at reducing the data dimensionality while preserving the initial interpretation of the variables, or features. This dimensionality reduction allows building simpler models, hence less prone to overfitting and local minima for instance, but it also brings extra information about the process being modeled.

As data are becoming higher- and higher-dimensional nowadays, feature selection methods need to adapt to cope with the computational burden and the problems induced by highly-correlated features. The mutual information, or the Gamma test, along with a greedy forward selection, is a very tractable option in such cases. Even though these criteria are much simpler than building a prediction model, like in a pure wrapper

approach, they can theoretically spot relationships of any kind between features and the variable to predict. Another challenge about feature selection methods is to assess their results through resampling methods, to make sure that the selected subset is not dependent on the actual sample used.

REFERENCES

Aha, D. W., & Bankert, R. L. (1996). A comparative evaluation of sequential feature selection algorithms. Chap. 4, pages 199–206 of: Fisher, Doug, & Lenz, Hans-J. (eds), *Learning from data : AI and statistics* V. Springer-Verlag.

Battiti, R. (1994). Using the mutual information for selecting features in supervised neural net learning. *IEEE transactions on neural networks*, 5, 537–550.

Blum, A., & Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial intelligence*, 97(1-2), 245–271.

Breiman, Leo (2001). Random Forests. *Machine Learning* 45 (1), 5-32

Brooks, S, Friel, N. & King, R. (2003). Classical model selection via simulated annealing. *Journal of the royal statistical society: Series b (statistical methodology)*, 65(2), 503–520.

Caruana, R., de Sa, V. R.. (2003). Benefitting from the variables that variable selection discards. *Journal of machine learning research*, 3, 1245–1264.

Dash, M., & Liu, H. (1997) Feature selection for classification. *Intelligent data analysis*, 1(3), 191–156.

Efron, B. Hastie, T, Johnstone, I. & Tibshirani, R. (2004). Least angle regression. *Annals of statistics*. 32(2):407-451

François, D., Rossi. F., Wertz, V. & Verleysen, M. (2007). Resampling methods for parameter-free and robust feature selection with mutual information, *Neurocomputing*, , 70(7-9), 1265-1275.

Gunter, S. & Bunke, H. (2002). Creation of classifier ensembles for handwritten word recognition using feature selection algorithms. *Proceedings of the eighth international workshop on frontiers in handwriting recognition, IEEE Computer Society.* 183

Guyon, I. & Elisseeff, A. (2006). *Feature extraction*. Berlin : Springer

Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial intelligence*, 97(1-2), 273–324.

Kira, K. & Rendell, L. (1992). A practical approach to feature selection. *Proceedeings of the International Conference on Machine Learning. (Aberdeen, July 1992), Sleeman, D. & Edwards, P. (eds).*Morgan Kaufmann, 249-256

Kraskov, A., Stögbauer, Harald, & Grassberger, P. (2004). Estimating mutual information. *Physical review* E, 69, 066138. Website: http://www.klab.caltech.edu/~kraskov/MILCA/

Krier, C, Rossi, F., François, D. Verleysen, M. (Forthcoming 2008) A data-driven functional projection approach for the selection of feature ranges in spectra with ICA or cluster analysis. Accepted for publication in *Chemometrics and Intelligent Laboratory Systems.*

Lee, J. A. & Verleysen, M. (2007). *Nonlinear dimensionality reduction.* Berlin: Springer.

Liu, H. and Motoda, H. (2007) *Computational Methods of Feature Selection* (Chapman & Hall/Crc Data Mining and Knowledge Discovery Series). Chapman & Hall/CRC.

Russell, S. J., & Norvig., P. 2003. *Artifical intelligence: A modern approach.* 2nd edn. Upper Saddle River: Prentice Hall.

Stefansson, A., Koncar, N., & Jones, A. J. (1997). A note on the gamma test. *Neural computing & applications*, 5(3), 131–133. Website: http://users.cs.cf.ac.uk/Antonia.J.Jones/GammaArchive/IndexPage.htm

Yang, J., & Honavar, V. G. (1998). Feature subset selection using a genetic algorithm. *IEEE intelligent systems*, 13(2), 44–49.

KEY TERMS

Embedded Method: Method for feature selection that consists in analyzing the structure of a prediction model (usually a regression of classification tree) to identify features that are important.

Feature: Attribute describing the data. For instance, in data describing a human body, features could be length, weight, hair color, etc. Is synonym with 'input', 'variable', "attribute", 'descriptor', among others.

Feature Extraction: Feature extraction is the process of extracting a reduced set of features from a larger one (dimensionality reduction). Feature extraction can be performed by feature selection, linear or nonlinear projections, or by ad-hoc techniques, in the case of image data for instance.

Feature Selection: Process of reducing the dimensionality of the data by selecting a relevant subset of the original features and discarding the others.

Filter: Method for feature selection that is applied before any prediction model is built.

Gamma Test: Method for estimating the variance of the noise affecting the data. This estimator is non-parametric, and can be used for assessing the relevance of a feature subset.

Mutual Information: Information-theoretic measure that estimates the degree of independence between two random variables. This estimator can be used for assessing the relevance of a feature subset.

Nonlinear Projection: Method for dimensionality reduction by creating a new, smaller, set of features that are nonlinear combinations of the original features.

Wrapper: Method for feature selection that is based on the performances of a prediction model to evaluate the predictive power of a feature subset.

Financial Time Series Data Mining

Indranil Bose
The University of Hong Kong, Hong Kong

Alvin Chung Man Leung
The University of Hong Kong, Hong Kong

Yiu Ki Lau
The University of Hong Kong, Hong Kong

INTRODUCTION

Movement of stocks in the financial market is a typical example of financial time series data. It is generally believed that past performance of a stock can indicate its future trend and so stock trend analysis is a popular activity in the financial community. In this chapter, we will explore the unique characteristics of financial time series data mining. Financial time series analysis came into being recently. Though the world's first stock exchange was established in the 18th century, stock trend analysis began only in the late 20th century. According to Tay et al. (2003) analysis of financial time series has been formally addressed only since 1980s.

It is believed that financial time series data can speak for itself. By analyzing the data, one can understand the volatility, seasonal effects, liquidity, and price response and hence predict the movement of a stock. For example, the continuous downward movement of the S&P index during a short period of time allows investors to anticipate that majority of stocks will go down in immediate future. On the other hand, a sharp increase in interest rate makes investors speculate that a decrease in overall bond price will occur. Such conclusions can only be drawn after a detailed analysis of the historic stock data. There are many charts and figures related to stock index movements, change of exchange rates, and variations of bond prices, which can be encountered everyday. An example of such a financial time series data is shown in Figure 1. It is generally believed that through data analysis, analysts can exploit the temporal dependencies both in the deterministic (regression) and the stochastic (error) components of a model and can come up with better prediction models for future stock prices (Congdon, 2003).

BACKGROUND

Financial time series are a sequence of financial data obtained in a fixed period of time. In the past, due to technological limitations, data was recorded on a weekly basis. Nowadays, data can be gathered for very short durations of time. Therefore, this data is also called high frequency data or tick by tick data. Financial time series data can be decomposed into several components. Kovalerchuk and Vityaev (2005) defined financial time series data as the summation of long term trends, cyclical variations, seasonal variations, and irregular movements. These special components make financial time series data different from other statistical data like population census that represents the growth trends in the population.

In order to analyze complicated financial time series data, it is necessary to adopt data mining techniques.

Figure 1. A typical movement of a stock index

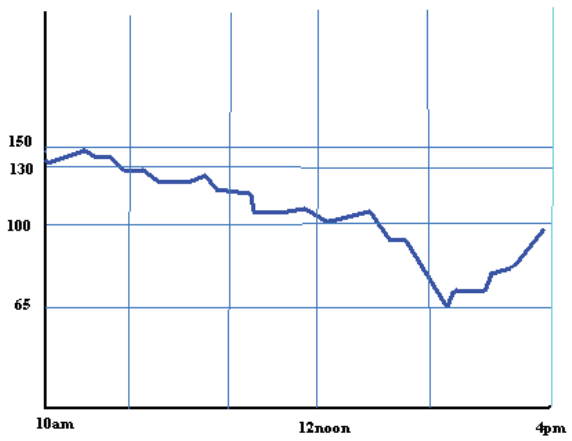


Table 1. Comparison of statistical and machine learning techniques

	Statistical techniques	Machine learning techniques
Advantages	• Relatively simple to use • Uni-variate and multi-variate analysis enable use of stationary and non-stationary models • Users can select different models to fit data and estimate parameters of models	• Easy to build model based on existing data • Computation carried out in parallel to model building which allows real time operations • Able to create own information representation during learning stages • More tolerant to noise in data
Disadvantages	• Performance and accuracy are negatively influenced by noise and non-linear components • The assumption of repeat patterns is unrealistic and may cause large errors in prediction	• Unstable for very large problems • Black box functions often do not provide any explanation of derived results

Currently, the commonly used data mining techniques are either statistics based or machine learning based. Table 1 compares the two types of techniques.

In the early days of computing, statistical models were popular tools for financial forecasting. According to Tsay (2002), the statistical models were used to solve linear time series problems, especially the stationarity and correlation problems of data. Models such as linear regression, autoregressive model, moving average, and autoregressive moving average dominated the industry for decades. However, those models were simple in nature and suffered from several shortcomings. Since financial data is rarely linear, these models were only useful to a limited extent. As a result sophisticated non-linear models like bilinear, threshold autoregression, smoothing transition autoregression, and conditional heteroscedastic autoregressions were developed to address the non-linearity in data. Those models could meet user requirements to a great extent. However, they were restricted in terms of the assumptions made by the models and were severely affected by the presence of noise in the data. It was observed that the correlation coefficient of diversification of a stock portfolio was adversely affected by the linear dependency of time series data (Kantardzic et al., 2004).

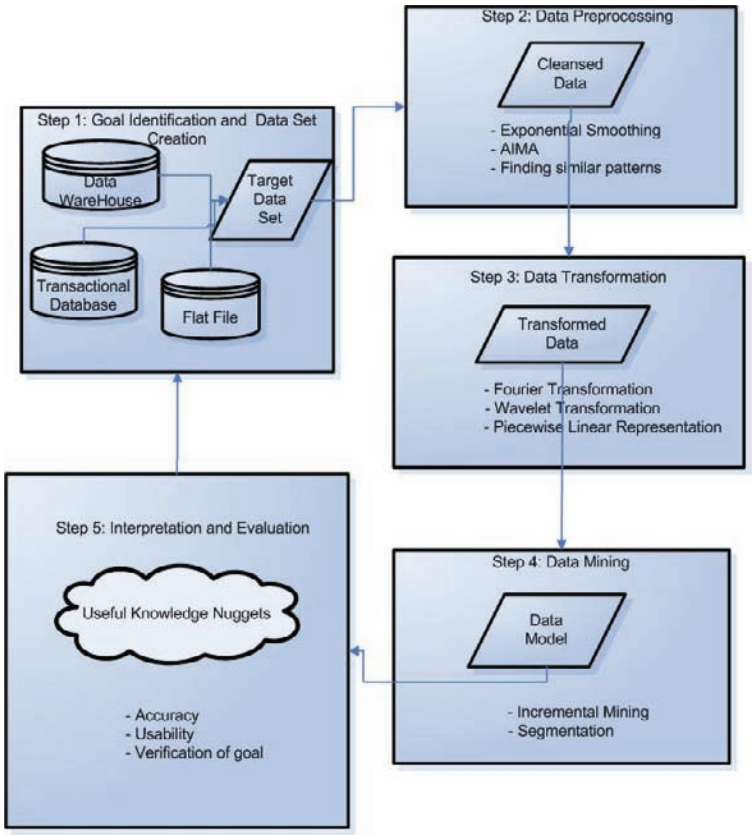
In the early 1990s, with the advent of machine learning new techniques were adopted by the field of financial forecasting. The machine learning techniques included artificial neural networks (ANN), genetic algorithms, decision trees, among others. ANNs soon became a very popular technique for predicting options prices, foreign exchange rates, stock and commodity prices, mutual funds, interest rates, treasury bonds etc. (Shen et al., 2005). The attractive features of ANNs were that it was data-driven, did not make any prior assumptions about the data, and could be used for linear and non-linear analysis. Also its tolerance to noise was high.

FINANCIAL TIME SERIES DATA MINING

In order to make useful predictions about future trends of financial time series, it is necessary to follow a number of steps of Knowledge Discovery in Databases (KDD) as shown in Figure 2. We can classify KDD into four important steps, namely, goal identification, preprocessing, data mining, and post-processing (Last et al., 2001). In order to conduct KDD, analysts have to establish a clear goal of KDD and prepare necessary data sets. Secondly, they have to preprocess the data to eliminate noise. Thirdly, it is necessary to transform the format of the data so as to make it amenable for data analysis. In the step of data mining, analysts have to supply a series of training data to build up a model, which can be subsequently tested using testing data. If the error of prediction for the training data does not exceed the tolerance level, the model will be deployed for use. Otherwise, the KDD process will be redone.

Step 1: Goal identification and data creation. To begin with KDD, users need to establish a clear goal

Figure 2. Steps involved in KDD



that serves as the guiding force for the subsequent steps. For example, users may set up a goal to find out which time period is best to buy or sell a stock. In order to achieve this goal, users have to prepare relevant data and this can include historical data of stocks. There are many financial time series data readily available for sale from vendors such as Reuters and Bloomberg. Users may purchase such data and analyze them in subsequent steps.

Step 2: Data preprocessing. Time series data can be classified into five main categories, namely, seasonal, trended, seasonal and trended, stationary, and chaotic (Cortez et al., 2001). If the noisy components in the data are not preprocessed, data mining cannot be conducted. De-noising is an important step in data preprocessing. Yu et al. (2006) showed that preprocessed data can increase the speed of learning for ANN. Several de-noising techniques are listed below.

Exponential smoothing: It is one of the most popular de-noising techniques used for time series data. By averaging historical values, it can identify random noise and separate it to make the data pattern consistent (Cortez et al., 2001). It is easy to use and is accurate for short term forecasting. It is able to isolate seasonal effects, shocks, psychological conditions like panic, and also imitation effects.

Autoregressive integrated moving average (ARIMA): Based on the linear combination of past values and errors, this technique makes use of the least squares method to identify noise in data (Cortez et al., 2001). Though this method is complex, it exhibits high accuracy of prediction over a wide range of time series domains.

Finding similar patterns: Apart from removing inconsistent data from databases, another method to generate clean time series data is by finding similar

patterns. Several techniques like longest common subsequence, Euclidean and time warping distance functions, and indexing techniques such as nearest neighbor classification are used for this purpose. Making use of estimation, the techniques mentioned above find similar patterns and can be used to facilitate pruning of irrelevant data from a given dataset. These techniques have shown their usefulness in discovering stocks with similar price movements (Vlachos et al., 2004).

Step 3: Data transformation. In order to provide a strong foundation for subsequent data mining, time series data needs to be transformed into other formats (Lerner et al., 2004). The data to be transformed include daily transaction volumes, turnovers, percentage change of stock prices, etc. Using data transformation, a continuous time series can be discretized into meaningful subsequences and represented using real-valued functions. (Chung et al., 2004) Three transformation techniques are frequently used and they are Fourier transformation, wavelet transformation, and piecewise linear representation.

Fourier Transformation: This method decomposes the input values into harmonic wave forms by using sine or cosine functions. It guarantees generation of a continuous and finite dataset.

Wavelet Transformation: This is similar to the above technique. Wavelet is a powerful data reduction technique that allows prompt similarity search over high dimensional time series data (Popivanov and Miller, 2002).

Piecewise Linear Representation: It refers to the approximation of time series 'T', of length 'n' with 'K' straight lines. Such a representation can make the storage, transmission and computation of data more efficient (Keogh et al., 2004).

Step 4: Data mining. Time series data can be mined using traditional data mining techniques such as ANN and other machine learning methods (Haykin, 1998). An important method of mining time series data is incremental learning. As time series data is on-going data, the previously held assumptions might be invalidated by the availability of more data in future. As the volume of data available for time series analysis is huge, re-mining is inefficient. Hence incremental learning to detect changes in an evaluated model is a useful method for time series data mining. There are many relevant methods for incremental mining, and they include induction of decision trees, semi-incremental learning methods, feed forward neural networks (Zeira et al., 2004) and also incremental update of specialized binary trees (Fu et al., 2005).

After incremental mining, time series data needs to be segmented into several categories. For example, highly volatile stocks and stocks that are responsive to movements of a particular index can be grouped into one category. There are three types of algorithms which help in such categorization. They are sliding windows, top-down, and bottom-up algorithms.

Sliding windows: Using this technique, an initial segment of the time series is first developed. Then it is grown incrementally till it exceeds some error bound. A full segment results when the entire financial time series data is examined (Keogh et al., 2004). The advantage of this technique is that it is simple and allows examination of data in real time. However, its drawbacks are its inability to look ahead and its lack of global view when compared to other methods.

Top-down: Unlike the previous technique, top-down algorithm considers every possible partitioning of time series. Time series data is partitioned recursively until some stopping criteria are met. This technique gives the most reasonable result for segmentation but suffers from problems related to scalability.

Bottom-up: This technique complements the top-down algorithm. It begins by first creating natural approximations of time series, and then calculating the cost of merging neighboring segments. It merges the lowest cost pair and continues to do so until a stopping criterion is met (Keogh et al., 2004). Like the previous technique, bottom-up is only suitable for offline datasets.

Step 5: Interpretation and evaluation. Data mining may lead to discovery of a number of segments in data. Next, the users have to analyze those segments in order to discover information that is useful and relevant for them. If the user's goal is to find out the most appropriate time to buy or sell stocks, the user must discover the relationship between stock index and

period of time. As a result of segmentation, the user may observe that price of a stock index for a particular segment of the market is extremely high in the first week of October whereas the price of another segment is extremely low in the last week of December. From that information, the user should logically interpret that buying the stock in the last week of December and selling it in the first week of October will be most profitable. A good speculation strategy can be formulated based on the results of this type of analysis. However, if the resulting segments give inconsistent information, the users may have to reset their goal based on the special characteristics demonstrated for some segments and then reiterate the process of knowledge discovery.

FUTURE TRENDS

Although statistical analysis plays a significant role in the financial industry, machine learning based methods are making big inroads into the financial world for time series data mining. An important development in time series data mining involves a combination of the sliding windows and the bottom-up segmentation. Keogh et al. (2004) have proposed a new algorithm by combining the two. It is claimed that the combination of these two techniques can reduce the runtime while retaining the merits of both methods.

Among machine learning techniques, ANNs seem to be very promising. However, in the recent literature, soft computing tools such as fuzzy logic, and rough sets have proved to be more flexible in handling real life situations in pattern recognition. (Pal & Mitra, 2004). Some researchers have even used a combination of soft computing approaches with traditional machine learning approaches. Examples of such include Neuro-fuzzy computing, Rough-fuzzy clustering and Rough self-organizing map (RoughSOM). Table 2 compares these three hybrid methods.

CONCLUSION

Financial time series data mining is a promising area of research. In this chapter we have described the main characteristics of financial time series data mining from the perspective of knowledge discovery in databases and we have elaborated the various steps involved in financial time series data mining. Financial time series data is quite different from conventional data in terms of data structure and time dependency. Therefore, traditional data mining processes may not be adequate for mining financial time series data. Among the two main streams of data analysis techniques, statistical models seem to be the more dominant due to their ease of use. However, as computation becomes cheaper and more data becomes available for analysis, machine learning techniques that overcome the inadequacies of statistical models are likely to make big inroads into the financial market. Machine learning tools such as

Table 2. Examples of hybrid soft computing methods for mining financial time series

New methods	Neuro-fuzzy computing	Rough-fuzzy clustering	RoughSOM
Description	An intelligent computing mechanism that integrates ANN with fuzzy logic.	A method which defines clusters using rough set theory and represents member values using fuzzy logic (Asharaf & Murty, 2004).	An SOM which classifies a new object by matching rules generated by rough sets (Shen & Loh, 2004)
Mechanism	Combination of neural network and fuzzy logic	Combination of rough sets and fuzzy logic	Combination of rough sets and self-organization maps
Advantages	Retained the advantage of ANN such as massive parallelism, robustness, and learning in data-rich environments and made modelling of imprecise and qualitative knowledge possible by using fuzzy logic (Pal & Mitra, 2004)	Proved to be scalable to large data sets and offered a faster and more robust solution to initialization and local minima problem of iterative refinement clustering (Pal & Mitra, 2004)	Took less initialization time and learning time and offered better cluster quality and more compact data representation (Pal & Mitra, 2004)

ANN as well as several soft computing approaches are gaining increasing popularity in this area due to their ease of use and high quality of performance. In future, it is anticipated that machine learning based methods will overcome statistical methods in popularity for time series analysis activities in the financial world.

REFERENCES

Asharaf, S. & Murty, M.N. (2004). A rough fuzzy approach to web usage categorization, *Fuzzy Sets and Systems*, 148(1), 119-129.

Congdon, P. (2003). *Applied Bayesian Modeling.* Chichester, West Sussex, England: **Wiley.**

Cortez, P., Rocha, M., & Neves, J. (2001). Evolving time series forecasting neural network models. In *Proceedings of the Third International Symposium on Adaptive Systems, Evolutionary Computation, and Probabilistic Graphical Models* (pp. 84-91). Havana, Cuba.

Chung, F.L., Fu, T.C., Ng, V. & Luk, R.W.P. (2004). An evolutionary approach to pattern-based time series segmentation, *IEEE Transactions on Evolutionary Computation*, 8(5), 471-489.

Fu, T., Chung, F., Tang, P., Luk, R., & Ng, C. (2005). Incremental stock time series data delivery and visualization. In *Proceedings of the Fourteenth ACM International Conference on Information and Knowledge Management* (pp. 279-280). New York, USA: ACM Press.

Haykin, S. (1998). Neural Networks: A Comprehensive Foundation Second Edition. New York, USA: Prentice Hall.

Kantardzic, M., Sadeghian, P., & Shen, C. (2004). The time diversification monitoring of a stock portfolio: An approach based on the fractal dimension. In *Proceedings of the 2004 ACM Symposium on Applied Computing* (pp. 637-641). New York, USA: ACM Press.

Keogh, E., Chu, S., Hart, D., & Pazzani, M. (2004). Segmenting time series: a survey and novel approach. In M. Last, A. Kandel, & H. Bunke (Eds.), *Data Mining in Time Series Database (Series in Machine Perception and Artificial Intelligence Volume 57),* (pp. 1-21). New Jersey, NY: World Scientific.

Kovalerchuk, B. & Vityaev, E. (2005). Data mining for financial applications. In O. Maimon, & L. Rokach (Eds.), *The Data Mining and Knowledge Discovery Handbook* (pp. 1203-1224). Berlin, Germany: Springer.

Last, M., Klein, Y. & Kandel, A. (2001). Knowledge discovery in time series databases, IEEE Transactions on Systems, Man and Cybernetics, Part B, 3(1), 160-169.

Lerner, A., Shasha, D., Wang, Z., Zhao, X. & Zhu, Y. (2004). Fast algorithms for time series with applications to finance, physics, music, biology and other suspects. In *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data* (pp. 965-968). New York, USA: ACM Press.

Pal, S.K. & Mitra, P. (2004). *Pattern Recognition Algorithms for Data Mining: Scalability, Knowledge Discovery and Soft Granular Computing***.** Boca Raton, USA: CRC Press.

Popivanov, I., & Miller, R.J. (2002). Similarity search over time series data using wavelets. In *Proceedings of the Eighteenth International Conference on Data Engineering* (pp. 212-221). New York: IEEE Press.

Shen, L. & Loh, H.T. (2004). Applying rough sets to market timing decisions, *Decision Support Systems*, 37(4), 583-597.

Shen, H., Xu, C., Han, X., & Pan, Y. (2005). Stock tracking: A new multi-dimensional stock forecasting approach. In *Proceedings of the Seventh International Conference on Information Fusion, Vol. 2,* (pp. 1375-1382). New York: IEEE Press.

Tay, F.E.H., Shen, L., & Cao, L. (2003). *Ordinary Shares, Exotic Methods: Financial Forecasting Using Data Mining Techniques***.** New Jersey, USA: World Scientific.

Tsay, R.S. (2002). *Analysis of Financial Time Series***.** New York, USA: Wiley.

Vlachos, M., Gunopulos, D., & Das, G. (2004). Indexing time series under conditions of noise. In M. Last, A. Kandel, & H. Bunke (Eds.), *Data Mining in Time Series Database (Series in Machine Perception and Artificial Intelligence Volume 57)* (pp. 67-100). New Jersey, USA: World Scientific.

Yu, L., Wang, S. & Lai, K.K. (2006). An integrated data preparation scheme for neural network data analysis, IEEE Transactions on Knowledge and Data Engineering, 18(2), 217-230.

Zeira, G., Maimon, O., Last, M., & Rokach, L. (2004). Change detection in classification models induced from time series data. In M. Last, A. Kandel, & H. Bunke (Eds.), *Data Mining in Time Series Database (Series in Machine Perception and Artificial Intelligence Volume 57)* (pp. 101-125). New Jersey, USA: World Scientific.

KEY TERMS

Data Preprocessing: A procedure to make input data more suitable for data analysis so as to generate accurate results. Usual procedures for data preprocessing include de-noising and removing missing data.

Data Transformation: A procedure which transforms data from one format to another so that the latter format is more suitable for data mining.

Fuzzy Logic: A logic which models imprecise and qualitative knowledge in the form of degree of truth and allows objects to be members of overlapping classes.

Knowledge Discovery Process: A series of procedures which extract useful knowledge from a set of data.

Rough Set (RS): A representation of a conventional set using upper and lower approximation of the set and is to represent uncertainty or vagueness in the membership of objects to sets.

Self-Organizing Map (SOM): A single layer feedforward neural network, which is used for unsupervised learning or discovering similar clusters among data.

Soft Computing: A consortium of computational techniques which give the most optimal solution based on given tolerance level of imprecision and conventional logic system.

Time Series Segmentation: A process which classifies input time series data into different segments for the purpose of data classification.

Flexible Mining of Association Rules

Hong Shen
Japan Advanced Institute of Science and Technology, Japan

INTRODUCTION

The discovery of association rules showing conditions of data co-occurrence has attracted the most attention in data mining. An example of an association rule is the rule "the customer who bought bread and butter also bought milk," expressed by *T(bread; butter)→ T(milk)*.

Let $I = \{x_1, x_2, \ldots, x_m\}$ be a set of (data) items, called the domain; let D be a collection of records (transactions), where each record, T, has a unique identifier and contains a subset of items in I. We define *itemset* to be a set of items drawn from I and denote an itemset containing k items to be k-itemset. The support of itemset X, denoted by $\tilde{A}(X/D)$, is the ratio of the number of records (in D) containing X to the total number of records in D. An *association rule* is an implication rule $\Rightarrow Y$, where $X; \subseteq I$ and $X \bigcap Y=0$. The confidence of $\Rightarrow Y$ is the ratio of $\sigma(\bigcup Y/D)$ to $\sigma(X/D)$, indicating that the percentage of those containing X also contain Y. Based on the user-specified minimum support *(minsup)* and confidence *(minconf)*, the following statements are true: An itemset X is frequent if $\sigma(X/D) \geq$ minsup, and an association rule $\Rightarrow XY$ is strong i $\bigcup XY$ is frequent and $\frac{\sigma(X \bigcup Y/D)}{\sigma(X/Y)}$, minconf. The problem of mining association rules is to find all strong association rules, which can be divided into two subproblems:

1. Find all the frequent itemsets.
2. Generate all strong rules from all frequent itemsets.

Because the second subproblem is relatively straightforward — we can solve it by extracting every subset from an itemset and examining the ratio of its support; most of the previous studies (Agrawal, Imielinski, & Swami, 1993; Agrawal, Mannila, Srikant, Toivonen, & Verkamo, 1996; Park, Chen, & Yu, 1995; Savasere, Omiecinski, & Navathe, 1995) emphasized on developing efficient algorithms for the first subproblem.

This article introduces two important techniques for association rule mining: (a) finding N most frequent itemsets and (b) mining multiple-level association rules.

BACKGROUND

An association rule is called *binary association rule* if all items (attributes) in the rule have only two values: 1 (yes) or 0 (no). Mining binary association rules was the first proposed data mining task and was studied most intensively. Centralized on the *Apriori* approach (Agrawal et al., 1993), various algorithms were proposed (Savasere et al., 1995; Shen, 1999; Shen, Liang, & Ng, 1999; Srikant & Agrawal, 1996). Almost all the algorithms observe the downward property that all the subsets of a frequent itemset must also be frequent, with different pruning strategies to reduce the search space. Apriori works by finding frequent k-itemsets from frequent (k-1)-itemsets iteratively for k=1, 2, …, m-1.

Two alternative approaches, mining on domain partition (Shen, L., Shen, H., & Cheng, 1999) and mining based on knowledge network (Shen, 1999) were proposed. The first approach partitions items suitably into disjoint itemsets, and the second approach maps all records to individual items; both approaches aim to improve the bottleneck of Apriori that requires multiple phases of scans (read) on the database.

Finding all the association rules that satisfy minimal support and confidence is undesirable in many cases for a user's particular requirements. It is therefore necessary to mine association rules more flexibly according to the user's needs. Mining different sets of association rules of a small size for the purpose of predication and classification were proposed (Li, Shen, & Topor, 2001; Li, Shen, & Topor, 2002; Li, Shen, & Topor, 2004; Li, Topor, & Shen, 2002).

MAIN THRUST

Association rule mining can be carried out flexibly to suit different needs. We illustrate this by introducing important techniques to solve two interesting problems.

Finding *N* Most Frequent Itemsets

Give $\subseteq x, yI$, we say that x is greater than y, or y is less than x, if $\sigma(x/D) > \sigma(y/D)$. The largest itemset in D is the itemset that occurs most frequently in D. We want to find the N largest itemsets in D, where N is a user-specified number of interesting itemsets. Because users are usually interested in those itemsets with larger supports, finding N most frequent itemsets is significant, and its solution can be used to generate an appropriate number of interesting itemsets for mining association rules (Shen, L., Shen, H., Pritchard, & Topor, 1998).

We define the rank of itemset x, denoted by $\theta(x)$, as follows: $\theta(x) = |\{\sigma(y/D)>\sigma(x/D), \varnothing\subset y\subseteq I\}|+1$. Call x a winner if $\theta(x)\leq N$ and $\sigma(x/D)\geq 1$, which means that x is one of the N largest itemsets and it occurs in D at least once. We don't regard any itemset with support 0 as a winner, even if it is ranked below N, because we do not need to provide users with an itemset that doesn't occur in D at all.

Use W to denote the set of all winners and call the support of the smallest winner the *critical support*, denoted by *crisup*. Clearly, W exactly contains all itemsets with support exceeding crisup; we also have crisup$\geq$1. It is easy to see that $|W|$ may be different from N: If the number of all itemsets occurring in D is less than N, $|W|$ will be less than N; $|W|$ may also be greater than N, as different itemsets may have the same support. The problem of finding the N largest itemsets is to generate W.

Let x be an itemset. Use $P_k(x)$ to denote the set of all k-subsets (subsets with size k) of x. Use U_k to denote $P_1(I)\cup\ldots\cup P_k(I)$, the set of all itemsets with a size not greater than k. Thus, we introduce the k-rank of x, denoted by $\theta_k(x)$, as follows: $\theta_k(x) = |\{y|\{\sigma(y/D)>x/D), y\in U_k\}|+1$. Call x a k-winner if $\theta_k(x)\leq N$ and $\sigma(x/D)\geq 1$, which means that among all itemsets with a size not greater than k, x is one of the N largest itemsets and also occurs in D at least once. Use W_k to denote the set of all k-winners. We define *k-critical-support*, denoted by *k-crisup*, as follows: If $|W_k|<N$, then k-crisup is 1; otherwise, k-crisup is the support of the smallest k-winner. Clearly, W_k exactly contains all itemsets with a size not greater than k and support not less than k-crisup. We present some useful properties of the preceding concepts as follows.

Property: Let k and i be integers such that $1\leq k\leq k+i \leq |I|$.

(1)Given $x\in U_k$, *we have* $x\in W_k$ iff $\sigma(x/D)\geq k$-crisup.
(2) If $W_{k-1} = W_k$, then $W=W_k$
(3)$W_{k+i}\bigcap U_k\subseteq W_k$.
(4) $1\leq k$-crisup$\leq(k+i)$-crisup.

To find all the winners, the algorithm makes multiple passes over the data. In the first pass, we count the supports of all 1-itemsets, select the N largest ones from them to form W_1, and then use W_1 to generate potential 2-winners with size (2). Each subsequent pass k involves three steps: First, we count the support for potential k-winners with size k (called candidates) during the pass over D; then select the N largest ones from a pool precisely containing supports of all these candidates and all (k-1)-winners to form W_k; finally, use W_k to generate potential (k+1)-winners with size k+1, which will be used in the next pass. This process continues until we can't get any potential (k+1)-winners with size k+1, which implies that $W_k+1 = W_k$. From Property 2, we know that the last W_k exactly contains all winners.

We assume that M_k is the number of itemsets with support equal to k-crisup and a size not greater than k, where $1\leq k\leq |I|$, and M is the maximum of all $M_1\sim M_{|I|}$. Thus, we have $|W_k|=N+M_k-1<N+M$. It was shown that the time complexity of the algorithm is proportional to the number of all the candidates generated in the algorithm, which is $O(\log(N+M)*min\{N+M,|I|\}*(N+M))$ (Shen et al., 1998). Hence, the time complexity of the algorithm is polynomial for bounded N and M.

Mining Multiple-Level Association Rules

Although most previous research emphasized mining association rules at a single concept level (Agrawal et al., 1993; Agrawal et al., 1996; Park et al., 1995; Savasere et al., 1995; Srikant & Agrawal, 1996), some techniques were also proposed to mine rules at generalized abstract (multiple) levels (Han & Fu, 1995). However, they can only find multiple-level rules in a fixed concept hierarchy. Our study in this fold is motivated by the goal of

mining multiple-level rules in all concept hierarchies (Shen, L., & Shen, H., 1998).

A concept hierarchy can be defined on a set of database attribute domains such as $D(a_1),\ldots,D(a_n)$, where for $i \in [1, n]$, a_i denotes an attribute, and $D(a_i)$ denotes the domain of a_i. The concept hierarchy is usually partially ordered according to a general-to-specific ordering. The most general concept is the null description *ANY,* whereas the most specific concepts correspond to the specific attribute values in the database. Given a set of $D(a_1),\ldots,D(a_n)$, we define a concept hierarchy H as follows: $H^n \rightarrow H^{n-1} \rightarrow \ldots \rightarrow H^0$, where $H^i = D(a_1^i) \times \cdots \times D(a_i^i)$ for $i \in [0,n]$, and $\{a_1,..., a_n\} = \{a_1^n,..., a_n^n\} \supset \{a_1^n,..., a_{n-1}^{n-1}\} \supset \ldots \supset \emptyset$. Here, H^n represents the set of concepts at the primitive level, H^{n-1} represents the concepts at one level higher than those at H^n, and so forth; H^0, the highest level hierarchy, may contain solely the most general concept, ANY. We also use $\{a_1^n,..., a_n^n\} \rightarrow \{a_1^n,..., a_{n-1}^{n-1}\} \rightarrow \ldots \rightarrow \{a_1^1\}$ to denote H directly, and H^0 may be omitted here.

We introduce FML items to represent concepts at any level of a hierarchy. Let *, called a *trivial digit,* be a "don't-care" digit. An *FML item* is represented by a sequence of digits, $x = x_1x_2 \ldots x_n$, $x_i \in D(a_i) \cup \{*\}$. The flat-set of x is defined as $S_f(x) = \{(I,x_i) \mid i \in [1; n]$ and $x_i \neq *\}$. Given two items x and y, x is called a generalized item of y if $S_f(x) \subseteq S_f(y)$, which means that x represents a higher-level concept that contains the lower-level concept represented by y. Thus, *5* is a generalized item of 35* due to $S_f(*5*) = \{(2,5)\} \subseteq S_f(35*) = \{(1,3),(2,5)\}$. If $S_f(x) = \emptyset$, then x is called a *trivial item,* which represents the most general concept, ANY.

Let T be an encoded transaction table, t a transaction in T, x an item, and c an itemset. We can say that (a) t supports x if an item y exists in t such that x is a generalized item of y and (b) t supports c if t supports every item in c. The support of an itemset c in T, $\sigma(c/T)$, is the ratio of the number of transactions (in T) that support c to the total number of transactions in T. Given a minsup, an itemset c is large if $\sigma(c/T)e \geq$ minsup; otherwise, it is small.

Given an itemset c, we define its simplest form as $F_s(c) = \{x \in c \mid \forall\, y \in c\ S_f(x) \not\subset S_f(y)\}$ and its complete form as $F_c(c) = \{x \mid S_f(x) \subseteq S_f(y),\ y \in c\}$.

Given an itemset c, we call the number of elements in $F_s(c)$ its *size* and the number of elements in $F_c(c)$ its *weight.* An itemset of size j and weight k is called a *(j)*-itemset, [k]-itemset, or *(j)*[k]-itemset. Let c be a *(j)*[k]-itemset. Use $G_i(c)$ to indicate the set of all [i]-generalized-subsets of c, where $i \leq k$. Thus, the set of all [k-1]-generalized-subsets of c can be generated as follows: $G_{k-1}(c) = \{F_s(F_c(c) - \in x\}) \mid xF_s(c)\}$; the size of $G_{k-1}(c)$ is j. Hence, for $k \geq 1$, c is a (1)-itemset iff $|G_{k-1}(c)| = 1$. With this observation, we call a [k-1]-itemset a self-extensive if a (1)[k]-itemset b exists such that $G_{k-1}(b) = \{a\}$; at the same time, we call b the extension result of a. Thus, all self-extensive itemsets as can be generated from all (1)-itemsets bs as follows: $F_c(a) = F_c(b) - F_s(b)$.

Let b be a (1)-itemset and $F_s(b) = \{x\}$. From $|F_c(b)| = |\{v \mid S_f(y) \subseteq S_f(x)\}| = 2^{|S_f(x)|}$, we know that b is a (1)[$2^{|S_f(x)|}$]-itemset. Let a be the self-extensive itemset generated by b; that is, a is a [$2^{|S_f(x)|}$-1]-itemset such that $F_c(a) = F_c(b) - F_s(b)$. Thus, if $|S_f(x)| > 1$, there exist y, $z \in F_s(a)$ and $y \neq z$ such that $S_f(x) = S_f(y) \cup S_f(z)$. Clearly, this property can be used to generate the corresponding extension result from any self-extensive [2^m-1]-itemset, where $m > 1$. For simplicity, given a self-extensive [2^m-1]-itemset a, where $m > 1$, we directly use $E_r(a)$ to denote its extension result. For example, $E_r(\{12*,1*3,*23\}) = \{123\}$.

The algorithm makes multiple passes over the database. In the first pass, we count the supports of all [2]-itemsets and then select all the large ones. In pass k-1, we start with L_{k-1}, the set of all large [k-1]-itemsets, and use L_{k-1} to generate C_k, a superset of all large [k]-itemsets. Call the elements in C_k *candidate itemsets,* and count the support for these itemsets during the pass over the data. At the end of the pass, we determine which of these itemsets are actually large and obtain L_k for the next pass. This process continues until no new large itemsets are found. Not that $L_1 = \{\{\text{trivial item}\}\}$, because {trivial item} is the unique [1]-itemset and is supported by all transactions.

The computation cost of the preceding algorithm, which finds all frequent FML itemsets is $O(\sum_{c \in C} s(x))$, where C is the set of all candidates, $g(c)$ is the cost for generating c as a candidate, and $s(c)$ is the cost for counting the support of c (Shen, L., & Shen, H., 1998). The algorithm is optimal if the method of support counting is optimal.

After all frequent FML itemsets have been found, we can proceed with the construction of strong FML rules. Use $r(l,a)$ to denote rule $F_s(a) \rightarrow F_s(F_c(l) - F_c(a))$, where l is an itemset and a l. Use $F_o(l,a)$ to denote $F_c(l) - F_c(a)$ and say that $F_o(l,a)$ is an outcome form of l or the outcome form of $r(l,a)$. Note that $F_o(l,a)$ represents only a specific form rather than a meaningful itemset, so it is not equivalent to any other itemset whose simplest

form is $F_s(F_o(l,a))$. Outcome forms are also called *outcomes* directly. Clearly, the corresponding relationship between rules and outcomes is one to one. An outcome is strong if it corresponds to a strong rule. Thus, all strong rules related to a large itemset can be obtained by finding all strong outcomes of this itemset.

Let l be an itemset. Use $O(l)$ to denote the set of all outcomes of l; that is, $O(l) = \{F_o(l,a) | a \quad l\}$. Thus, from $O(l)$, we can output all rules related to l: $F_s(F_c(l)\text{-}o) \Rightarrow F_s(o)$ (denoted by $\hat{r}(l,o)$), where $o O(l)$. Clearly, $r(la)$ and $\hat{r}(l,o)$ denote the same rule if $o=F_o(l,a)$. Let o, $\hat{o} \in O(l)$. We can say two things: (a) o is a $|k|$-outcome of l if o exactly contains k elements and (b) is a sub-outcome of o verus l if. Use $O_k(l)$ to denote the set of all the $|k|$-outcomes of l and use $V_m(o,l)$ to denote the set of all the $|m|$-sub-outcomes of o versus l. Let o be an $|m+1|$-outcome of l and $m e>1$. If $|V_m(o,l)| =1$, then o is called an elementary outcome; otherwise, o is a non-elementary outcome.

Let $r(l,a)$ and $r(l,b)$ be two rules. We can say that $r(l,a)$ is an instantiated rule of $r(lb)$ if $b \quad a$. Clearly, b a implies $\sigma(b/T) \geq \sigma(a/)$ and

$\frac{\sigma(l/T)}{\sigma(a/T)} \geq \frac{\sigma(l/T)}{\sigma(b/T)}$. Hence, all instantiated rules of a strong rule must also be strong. Let l be a large itemset $o_1, o_2 \in O(l)$, $o_1=F_o(l,a)$, and $o_2 = F_o(l,b)$. The three straightforward conclusions are (a) $o_1 \subseteq o_2$ iff $b \quad a$, (b) $\hat{r}(l,o)= r(l,a)$,and (c) $\hat{r}(l,o_2) = r(l,b)$. Therefore $\hat{r}(l,o_1)$,is an instantiated rule of $\hat{r}(l,o_2)$ iff $o_1 \subseteq o_2$, which implies that all the suboutcomes of a strong outcome must also be strong. This characteristic is similar to the property that all generalized subsets of a large itemset must also be large. Hence, the algorithm works as follows. From a large itemset l, we first generate all the strong rules with |1|-outcomes (ignore the |0|-outcome Ø, which only corresponds to all trivial strong rule; $Fs(l) \Rightarrow Ø$ for every large itemset l). We then use all these strong |1|-outcomes to generate all the possible strong |2|-outcomes of l, from which all the strong rules with |2|-outcomes can be produced, and so forth.

The computation cost of the algorithm constructing all the strong FML rules is $O(|C|t + |C|s)$ (Shen, L., & Shen, H., 1998), which is optimal for bounded costs t and s, where C is the set of all the candidate rules, s is the average cost for generating one element in C, and t is the average cost for computing the confidence of one element in C.

FUTURE TRENDS

The discovery of data association is a central topic of data mining. Mining this association flexibly to suit different needs has an increasing significance for applications. Future research trends include mining association rules in data of complex types such as multimedia, time-series, text, and biological databases, in which certain properties (e.g., multidimensionality), constraints (e.g., time variance), and structures (e.g., sequence) of the data must be taken into account during the mining process; mining association rules in databases containing incomplete, missing, and erroneous data, which requires people to produce correct results regardless of the unreliability of data; mining association rules online for streaming data that come from an open-ended data stream and flow away instantaneously at a high rate, which usually adopts techniques such as active learning, concept drift, and online sampling and proceeds in an incremental manner.

CONCLUSION

We have summarized research activities in mining association rules and some recent results of my research in different streams of this topic. We have introduced important techniques to solve two interesting problems of mining N most frequent itemsets and mining multiple-level association rules, respectively. These techniques show that mining association rules can be performed flexibly in user-specified forms to suit different needs. More relevant results can be found in Shen (1999). We hope that this article provides useful insight to researchers for better understanding to more complex problems and their solutions in this research direction.

REFERENCES

Agrawal, R., Imielinski, T., & Swami, A. (1993). Mining associations between sets of items in massive databases. *Proceedings of the ACM SIGMOD International Conference on Management of Data* (pp. 207-216).

Agrawal, R., Mannila, H., Srikant,Toivonen, R. H., & Verkamo, A. I. (1996). Fast discovery of association rules. In U. Fayyad (Ed.), *Advances in knowledge discovery and data mining* (pp. 307-328). MIT Press.

Han, J., & Fu, Y. (1995). Discovery of multiple-level association rules from large databases. *Proceedings of the 21st VLDB International Conference* (pp. 420-431).

Li, J., Shen, H., & Topor, R. (2001). Mining the smallest association rule set for predictions. *Proceedings of the IEEE International Conference on Data Mining* (pp. 361-368).

Li, J., Shen, H., & Topor, R. (2002). Mining the optimum class association rule set. *Knowledge-Based Systems, 15*(7), 399-405.

Li, J., Shen, H., & Topor, R. (2004). Mining the informative rule set for prediction. *Journal of Intelligent Information Systems, 22*(2), 155-174.

Li, J., Topor, R., & Shen, H. (2002). Construct robust rule sets for classification. *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining* (pp. 564-569).

Park, J. S., Chen, M., & Yu, P. S. (1995). An effective hash based algorithm for mining association rules. *Proceedings of the ACM SIGMOD International Conference on Management of Data* (pp. 175-186).

Savasere, A., Omiecinski, R., & Navathe, S. (1995). An efficient algorithm for mining association rules in large databases. *Proceedings of the 21st VLDB International Conference* (pp. 432-443).

Shen, H. (1999). New achievements in data mining. In J. Sun (Ed.), *Science: Advancing into the new millenium* (pp. 162-178). Beijing: People's Education.

Shen, H., Liang, W., & Ng, J. K-W. (1999). Efficient computation of frequent itemsets in a subcollection of multiple set families. *Informatica, 23*(4), 543-548.

Shen, L., & Shen, H. (1998). Mining flexible multiple-level association rules in all concept hierarchies. *Proceedings of the Ninth International Conference on Database and Expert Systems Applications* (pp. 786-795).

Shen, L., Shen, H., & Cheng, L. (1999). New algorithms for efficient mining of association rules. *Information Sciences, 118,* 251-268.

Shen, L., Shen, H., Pritchard, P., & Topor, R. (1998). Finding the *N* largest itemsets. *Proceedings of the IEEE International Conference on Data Mining, 19* (pp. 211-222).

Srikant, R., & Agrawal, R. (1996). Mining quantitative association rules in large relational tables. *ACM SIGMOD Record, 25*(2), 1-8.

KEY TERMS

Association Rule: An implicatio⇒ rule *XY* that shows the conditions of co-occurrence of disjoint itemsets (attribute value sets) *X* and *Y* in a given database.

Concept Hierarchy: The organization of a set of database attribute domains into different levels of abstraction according to a general-to-specific ordering.

Confidence o⇒ Rule *XY*: The fraction of the database containing *X* that also contains *Y,* which is the ratio of the supp⋃rt of *XY* to the support of *X*.

Flexible Mining of Association Rules: Mining association rules in user-specified forms to suit different needs, such as on dimension, level of abstraction, and interestingness.

Frequent Itemset: An itemset that has a support greater than user-specified minimum support.

Strong Rule: An association rule whose support (of the union of itemsets) and confidence are greater than user-specified minimum support and confidence, respectively.

Support of Itemset *X*: The fraction of the database that contains *X*, which is the ratio of the number of records containing *X* to the total number of records in the database.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 509-513, copyright 2005 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

Formal Concept Analysis Based Clustering

Jamil M. Saquer
Southwest Missouri State University, USA

INTRODUCTION

Formal concept analysis (FCA) is a branch of applied mathematics with roots in lattice theory (Wille, 1982; Ganter & Wille, 1999). It deals with the notion of a concept in a given universe, which it calls context. For example, consider the context of transactions at a grocery store where each transaction consists of the items bought together. A concept here is a pair of two sets (A, B). A is the set of transactions that contain all the items in B and B is the set of items common to all the transactions in A. A successful area of application for FCA has been data mining. In particular, techniques from FCA have been successfully used in the association mining problem and in clustering (Kryszkiewicz, 1998; Saquer, 2003; Zaki & Hsiao, 2002). In this article, we review the basic notions of FCA and show how they can be used in clustering.

BACKGROUND

A fundamental notion in FCA is that of a context, which is defined as a triple (G, M, I), where G is a set of objects, M is a set of features (or attributes), and I is a binary relation between G and M. For object g and feature m, gIm if and only if g possesses the feature m. An example of a context is given in *Table 1*, where an "X" is placed in the i^{th} row and j^{th} column to indicate that the object in row i possesses the feature in column j.

The set of features common to a set of objects A is denoted by $\beta(A)$ and is defined as $\{m \in M \mid gIm \ \forall\ g \in A\}$. Similarly, the set of objects possessing all the features in a set of features B is denoted by $\alpha(B)$ and is given by $\{g \in G \mid gIm \ \forall m \in B\}$. The operators α and β satisfy the assertions given in the following lemma.

Lemma 1 (Wille, 1982): Let (G, M, I) be a context. Then the following assertions hold:

1. $A_1 \subseteq A_2$ implies $\beta(A_2) \subseteq \beta(A_1)$ for every $A_1, A_2 \subseteq G$, and $B_1 \subseteq B_2$ implies $\alpha(B_2) \subseteq \alpha(B_1)$ for every $B_1, B_2 \subseteq M$.
2. $A \subseteq \alpha(\beta(A)$ and $A = \beta(\alpha(\beta(A))$ for all $A \subseteq G$, and $B \subseteq \beta(\alpha((B))$ and $B = \alpha(\beta(\alpha\ (B)))$ for all $B \subseteq M$.

Table 1. A context excerpted from (Ganter, and Wille, 1999, p. 18). a = needs water to live; b = lives in water; c = lives on land; d = needs chlorophyll; e = two seeds leaf; f = one seed leaf; g = can move around; h = has limbs; i = suckles its offsprings.

		a	b	c	d	e	f	g	h	i
1	Leech	X	X					X		
2	Bream	X	X					X	X	
3	Frog	X	X	X				X	X	
4	Dog	X		X				X	X	X
5	Spike-weed	X	X		X		X			
6	Reed	X	X	X	X		X			
7	Bean	X		X	X	X				
8	Maize	X		X	X		X			

A formal concept in the context (G, M, I) is defined as a pair (A, B) where $A \subseteq G$, $B \subseteq M$, $\beta(A) = B$, and $\alpha(B) = A$. A is called the extent of the formal concept and B is called its intent. For example, the pair (A, B) where A = {2, 3, 4} and B = {a, g, h} is a formal concept in the context given in *Table 1*. A subconcept/superconcept order relation on concepts is as follows: $(A_1, B_1) \leq (A_2, B_2)$ iff $A_1 \subseteq A_2$ (or equivalently, iff $B_2 \subseteq B_1$). The fundamental theorem of FCA states that the set of all concepts on a given context is a complete lattice, called the concept lattice (Ganter & Wille, 1999). Concept lattices are drawn using Hasse diagrams, where concepts are represented as nodes. An edge is drawn between concepts C_1 and C_2 iff $C_1 \leq C_2$ and there is no concept C_3 such that $C_1 \leq C_3 \leq C_2$. The concept lattice for the context in Table 1 is given in Figure 1.

A less condensed representation of a concept lattice is possible using reduced labeling (Ganter & Wille, 1999). Figure 2 shows the concept lattice in Figure 1 with reduced labeling. It is easier to see the relationships and similarities among objects when reduced labeling is used. The extent of a concept C in *Figure 2* consists of the objects at C and the objects at the concepts that can be reached from C going downward following descending paths towards the bottom concept. Similarly, the intent of C consists of the features at C and the features at the concepts that can be reached from C going upwards following ascending paths to the top concept.

Consider the context presented in *Table 1*. Let B = {a, f}. Then, $\alpha(B) = \{5, 6, 8\}$, and $\beta(\alpha(B)) = b(\{5, 6, 8\}) = \{a, d, f\} \neq \{a, f\}$; therefore, in general, $\beta(\alpha(B)) \neq B$. A set of features B that satisfies the condition $b(\alpha(B)) = B$ is called a closed feature set. Intuitively, a closed feature set is a maximal set of features shared by a set of objects. It is easy to show that intents of the concepts of a concept lattice are all closed feature sets.

The support of a set of features B is defined as the percentage of objects that possess every feature in B. That is, $support(B) = |\alpha(B)|/|G|$, where |B| is the

Figure 1. Concept lattice for the context in Table 1

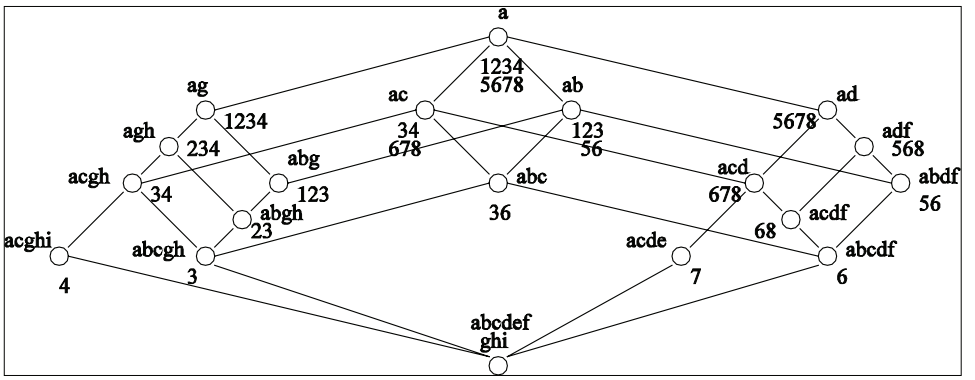
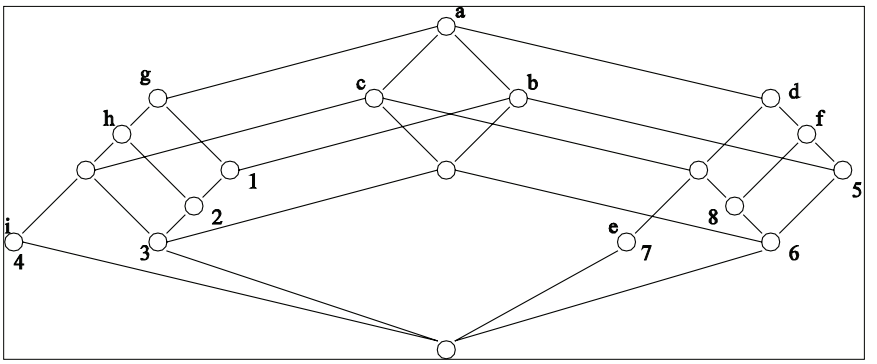


Figure 2. Concept lattice for the context in Table 1 with reduced labeling



cardinality of B. Let minSupport be a user-specified threshold value for minimum support. A feature set B is frequent iff support(B) ≥ minSupport. A frequent closed feature set is a closed feature set, which is also frequent. For example, for minSupport = 0.3, {a, f} is frequent, {a, d, f} is frequent closed, while {a, c, d, f} is closed but not frequent.

CLUSTERING BASED ON FCA

It is believed that the method described below is the first for using FCA for disjoint clustering. Using FCA for conceptual clustering to gain more information about data is discussed in Carpineto & Romano (1999) and Mineau & Godin (1995). In the remainder of this article we show how FCA can be used for clustering.

Traditionally, most clustering algorithms do not allow clusters to overlap. However, this is not a valid assumption for many applications. For example, in Web documents clustering, many documents have more than one topic and need to reside in more than one cluster (Beil, Ester, & Xu, 2002; Hearst, 1999; Zamir & Etzioni, 1998). Similarly, in the market basket data, items purchased in a transaction may belong to more than one category of items.

The concept lattice structure provides a hierarchical clustering of objects, where the extent of each node could be a cluster and the intent provides a description of that cluster. There are two main problems, though, that make it difficult to recognize the clusters to be used. First, not all objects are present at all levels of the lattice. Second, the presence of overlapping clusters at different levels is not acceptable for disjoint clustering. The techniques described in this chapter solve these problems. For example, for a node to be a cluster candidate, its intent must be frequent (meaning a minimum percentage of objects must possess all the features of the intent). The intuition is that the objects within a cluster must contain many features in common. Overlapping is resolved by using a score function that measures the goodness of a cluster for an object and keeps the object in the cluster where it scores best.

Formalizing the Clustering Problem

Given a set of objects $G = \{g_1, g_2, \ldots, g_n\}$, where each object is described by the set of features it possesses, (i.e., g_i is described by $\beta(g_i)$), $\mathbb{C} = \{C_1, C_2, \ldots, C_k\}$ is a clustering of G if and only if each C_i is a subset of G and $\bigcup C_i = G$, where i ranges from 1 to k. For disjoint clustering, the additional condition $C_i \bigcap C_j = \varnothing$ must be satisfied for all $i \neq j$.

Our method for disjoint clustering consists of two steps. First, assign objects to their initial clusters. Second, make these clusters disjoint. For overlapping clustering, only the first step is needed.

Assigning Objects to Initial Clusters

Each frequent closed feature set (FCFS) is a cluster candidate, with the FCFS serving as a label for that cluster. Each object g is assigned to the cluster candidates described by the maximal frequent closed feature set (MFCFS) contained in $\beta(g)$. These initial clusters may not be disjoint because an object may contain several MFCFS. For example, for minSupport = 0.375, object 2 in *Table 1* contains the MFCFSs agh and abg.

Notice that all of the objects in a cluster must contain all the features in the FCFS describing that cluster (which is also used as the cluster label). This is always true even after any overlapping is removed. This means that this method produces clusters with their descriptions. This is a desirable property. It helps a domain expert to assign labels and descriptions to clusters.

Making Clusters Disjoint

To make the clusters disjoint, we find the best cluster for each overlapping object g and keep g only in that cluster. To achieve this, a score function is used. The function score(g, C_i) measures the goodness of cluster C_i for object g. Intuitively, a cluster is good for an object g if g has many frequent features which are also frequent in C_i. On the other hand, C_i is not good for g if g has frequent features that are not frequent in C_i.

Define global-support(f) as the percentage of objects possessing f in the whole database, and cluster-support(f) as the percentages of objects possessing f in a given cluster C_i. We say f is cluster frequent in C_i if cluster-support(f) is at least a user-specified minimum threshold value q. For cluster C_i, let positive(g, C_i) be the set of features in $\beta(g)$ which are both global-frequent (i.e., frequent in the whole database) and cluster-frequent. Also let negative(g, C_i) be the set of features in $\beta(g)$ which are global-frequent but not cluster-frequent. The function score(g, C_i) is then given by the following formula:

Table 2. Initial cluster assignments and support values. (minSupport = 0.37, θ = 0.7)

C_i	initial clusters	Positive(g,C_i)	negative(g,C_i)
C[agh]	2: abgh 3: abcgh 4: acghi	a (1), g (1), h(1)	b (0.63), c(0.63)
C[abg]	1: abg 2: abgh 3: abcgh	a (1), b (1), g(1)	c(0.63), h(0.38)
C[acd]	6: abcdf 7: acde 8: acdf	a (1), c (1), d(1)	b (0.63), f(0.38)
C[adf]	5: abdf 6: abcdf 8: acdf	a (1), d(1), f(1)	b (0.63), c(0.63)

Table 3. Score calculations and final cluster assignments. (minSupport = 0.37, □ = 0.7)

C_i	Score(g, C_i)	Final clusters
C[agh]	2: 1 - .63 + 1 + 1 = 2.37 3: 1 - .63 - .63 + 1 + 1 = 1.74 4: does not overlap	 4
C[abg]	1: does not overlap 2: 1 + 1 + 1 - .38 = 2.62 3: 1 + 1 - .63 + 1 - .38 = 1.99	1 2 3
C[acd]	6: 1 - .63 + 1 +1 - .38 = 2.62 5: does not overlap 8: 1 + 1 + 1 -.38 = 2.62	6 7 8
C[adf]	5: does not overlap 6: 1 - .63 - .63 + 1 + 1 = 1.74 8: 1 - .63 + 1 + 1 = 2.37	 5

$$score(g, C_i) = \sum_{f \in positive(g, Ci)} cluster\text{-}support(f) - \sum_{f \in negative(g, Ci)} global\text{-}support(f).$$

The first term in score(g, C_i) favors C_i for every feature in positive(g, C_i) because these features contribute to intra-cluster similarity. The second term penalizes C_i for every feature in negative(g, C_i) because these features contribute to inter-cluster similarities.

An overlapping object will be deleted from all initial clusters except for the cluster where it scores highest. Ties are broken by assigning the object to the cluster with the longest label. If this does not resolve ties, then one of the clusters is chosen randomly.

An Illustrating Example

Consider the objects in *Table* 1. The closed feature sets are the intents of the concept lattice in *Figure 1*. For minSupport = 0.35, a feature set must appear in at least 3 objects to be frequent. The frequent closed feature

sets are a, ag, ac, ab, ad, agh, abg, acd, and adf. These are the candidate clusters. Using the notation C[x] to indicate the cluster with label x, and assigning objects to MFCFS results in the following initial clusters: C[agh] = {2, 3, 4}, C[abg] = {1, 2, 3}, C[acd] = {6, 7, 8}, and C[adf] = {5, 6, 8}. To find the most suitable cluster for object 6, we need to calculate its score in each cluster containing it. For cluster-support threshold value θ of 0.7, it is found that, score(6,C[acd]) = 1 - 0.63 + 1 + 1 - 0.38 = 1.99 and score[6,C[adf]) = 1 - 0.63 - 0.63 + 1 + 1 = 1.74.

We will use score(6,C[acd]) to explain the calculation. All features in $\beta(6)$ are global-frequent. They are a, b, c, d, and f, with global frequencies 1, 0.63, 0.63, 0.5, and 0.38, respectively. Their respective cluster-support values are 1, 0.33, 1, 1, and 0.67. For a feature to be cluster-frequent in C[acd], it must appear in at least [θ x |C[acd]|] = 3 of its objects. Therefore, a, c, d ∈ positive(6,C[acd]), and b, f ∈ negative(6,C[acd]). Substituting these values into the formula for the score function, it is found that the score(6,C[acd]) = 1.99. Since score(6,C[acd]) > score[6,C[adf]), object 6 is assigned to C[acd].

Tables 2 and *3* show the score calculations for all overlapping objects. *Table 2* shows initial cluster assignments. Features that are both global-frequent and cluster-frequent are shown in the column labeled positive(g,C_i), and features that are only global-frequent are in the column labeled negative(g,C_i). For elements in the column labeled positive(g,C_i), the cluster-support values are listed between parentheses after each feature name. The same format is used for global-support values for features in the column labeled negative(g,C_i). It is only a coincidence that in this example all cluster-support values are 1 (try θ = 0.5 for different values). Table 3 shows score calculations, and final cluster assignments.

Notice that, different threshold values may result in different clusters. The value of minSupport affects cluster labels and initial clusters while that of θ affects final elements in clusters. For example, for minSupport = 0.375 and θ = 0.5, we get the following final clusters C[agh] = {3, 4}, C[abg] = {1, 2}, C[acd] = {7, 8}, and C[adf] = {5, 6}.

FUTURE TRENDS

We have shown how FCA can be used for clustering. Researchers have also used the techniques of FCA and frequent itemsets in the association mining problem (Fung, Wang, & Ester, 2003; Wang, Xu, & Liu, 1999; Yun, Chuang, & Chen, 2001). We anticipate this framework to be suitable for other problems in data mining such as classification. Latest developments in efficient algorithms for generating frequent closed itemsets make this approach efficient for handling large amounts of data (Gouda & Zaki, 2001; Pei, Han, & Mao, 2000; Zaki & Hsiao, 2002).

CONCLUSION

This chapter introduces formal concept analysis (FCA); a useful framework for many applications in computer science. We also showed how the techniques of FCA can be used for clustering. A global support value is used to specify which concepts can be candidate clusters. A score function is then used to determine the best cluster for each object. This approach is appropriate for clustering categorical data, transaction data, text data, Web documents, and library documents. These data usually suffer from the problem of high dimensionality with only few items or keywords being available in each transaction or document. FCA contexts are suitable for representing this kind of data.

REFERENCES

Beil, F., Ester, M., & Xu, X. (2002). Frequent term-based text clustering. In *The 8th International Conference on Knowledge Discovery and Data Mining (KDD 2002)* (pp. 436-442).

Carpineto, C., & Romano, G. (1993). GALOIS: An order-theoretic approach to conceptual clustering. In *Proceedings of 1993 International Conference on Machine Learning* (pp. 33-40).

Fung, B., Wang, K., & Ester, M. (2003). Large hierarchical document clustering using frequent itemsets. In *Third SIAM International Conference on Data Mining* (pp. 59-70).

Ganter, B., & Wille, R. (1999). *Formal concept analysis: Mathematical foundations.* Berlin: Springer-Verlag.

Gouda, K., & Zaki, M. (2001). Efficiently mining maximal frequent itemsets. In *First IEEE International*

Conference on Data Mining (pp. 163-170). San Jose, USA.

Hearst, M. (1999). The use of categories and clusters for organizing retrieval results. In T. Strzalkowski (Ed.), *Natural language information retrieval* (pp. 333-369). Boston: Kluwer Academic Publishers.

Kryszkiewicz, M. (1998). Representative association rules. In *Proceedings of PAKDD '98. Lecture Notes in Artificial Intelligence* (Vol. 1394) (pp. 198-209). Berlin: Springer-Verlag.

Mineau, G., & Godin, R. (1995). Automatic structuring of knowledge bases by conceptual clustering. *IEEE Transactions on Knowledge and Data Engineering, 7* (5), 824-829.

Pei J., Han J., & Mao R. (2000). CLOSET: an efficient algorithm for mining frequent closed itemsets. In *ACM-SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery* (pp. 21-30). Dallas, USA.

Saquer, J. (2003). Using concept lattices for disjoint clustering. In *The Second IASTED International Conference on Information and Knowledge Sharing* (pp. 144-148).

Wang, K., Xu, C., & Liu, B. (1999). Clustering transactions using large items. In *ACM International Conference on Information and Knowledge Management* (pp. 483-490).

Wille, R. (1982). Restructuring lattice theory: An approach based on hierarchies of concepts. In I. Rival (Ed.), *Ordered sets* (pp. 445-470). Dordecht-Boston: Reidel.

Yun, C., Chuang, K., & Chen, M. (2001). An efficient clustering algorithm for market basket data based on small large ratios. In *The 25th COMPSAC Conference* (pp. 505-510).

Zaki, M.J., & Hsiao, C. (2002). CHARM: An efficient algorithm for closed itemset mining. In *Second SIAM International Conference on Data Mining.*

Zamir, O., & Etzioni, O. (1998). Web document clustering: A feasibility demonstration. In *The 21st Annual International ACM SIGIR* (pp. 46-54).

KEY TERMS

Cluster-Support of Feature F In Cluster C_i: Percentage of objects in C_i possessing f.

Concept: A pair (A, B) of a set A of objects and a set B of features such that B is the maximal set of features possessed by all the objects in A and A is the maximal set of objects that possess every feature in B.

Context: A triple (G, M, I) where G is a set of objects, M is a set of features and I is a binary relation between G and M such that gIm if and only if object g possesses the feature m.

Formal Concept Analysis: A mathematical framework that provides formal and mathematical treatment of the notion of a concept in a given universe.

Negative(g, Ci): Set of features possessed by g which are global-frequent but not cluster-frequent.

Positive(g, Ci): Set of features possessed by g which are both global-frequent and cluster-frequent.

Support or Global-Support of Feature F: Percentage of object "transactions" in the whole context (or whole database) that possess f.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 514-518, copyright 2005 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

Frequent Sets Mining in Data Stream Environments

Xuan Hong Dang
Nanyang Technological University, Singapore

Wee-Keong Ng
Nanyang Technological University, Singapore

Kok-Leong Ong
Deakin University, Australia

Vincent Lee
Monash University, Australia

INTRODUCTION

In recent years, data streams have emerged as a new data type that has attracted much attention from the data mining community. They arise naturally in a number of applications (Brian et al., 2002), including financial service (stock ticker, financial monitoring), sensor networks (earth sensing satellites, astronomic observations), web tracking and personalization (web-click streams). These stream applications share three distinguishing characteristics that limit the applicability of most traditional mining algorithms (Minos et al., 2002; Pedro and Geoff, 2001): (1) the continuous arrival rate of the stream is high and unpredictable; (2) the volume of data is unbounded, making it impractical to store the entire content of the stream; (3) in terms of practical applicability, stream mining results are often expected to be closely approximated the exact results as well as to be available at any time. Consequently, the main challenge in mining data streams is to develop effective algorithms that support the processing of stream data in one-pass manner (preferably on-line) whilst operating under system resources limitations (e.g., memory space, CPU cycles or bandwidth).

This chapter discusses the above challenge in the context of finding frequent sets from transactional data streams. The problems will be presented and some effective methods, both from deterministic and probabilistic approaches, are reviewed in details. The trade-offs between memory space and accuracy of mining results are also discussed. Furthermore, the problems will be considered in three fundamental mining models for stream environments: landmark window, forgetful window and sliding window models.

BACKGROUND

Generally, the problem of finding frequent sets from a data stream can be stated as follows. Let $I = \{a_1, a_2,... a_m\}$ be a set of items (or objects) and let X be an itemset such as $X \subseteq I$. Given a transactional data stream, DS, which is a sequence of incoming transactions, $(t_1, t_2,...,t_n)$, where $t_i \subseteq I$, the frequency of X is defined as the number of transactions in DS that contain X and its support is defined as the ratio between its frequency and the number of transactions in the stream. Then, the problem of mining frequent sets from data stream DS is to find all itemsets whose support is greater than a given threshold $s \in (0,1)$ called minimum support.

However, due to the unlimited size of streaming data, all transactions appearing in the stream DS are not always of equal importance. Rather, their usefulness is dependent upon applications. For example, some applications focus only on a fixed portion (most recently arriving) of the stream, while other applications consider all transactions seen so far; yet the transaction weight (or importance) is reduced gradually with time. This gives rise to different mining models: Landmark window model, Forgetful window model and Sliding window model. In the following section, most typical algorithms classified in these models will be discussed in details.

MAIN FOCUS

This section presents and discusses some typical algorithms addressing the problem of finding frequent sets from data streams. The mining model is first described, following it are the algorithm analysis and discussion.

Landmark Window Model

In this model, frequent sets are computed from a set of contiguous transactions in the stream identified between a specific transaction in the past, called landmark, and the current transaction. Accordingly, one endpoint (landmark) is fixed, while the other gradually increases with the arrival of new transactions. This model is often suitable for off-line data streams where transactions usually arrive in batches (Gurmeet and Rajeev, 2002), for example, with warehouse applications where new batches of records are updated at regular time intervals. However, there are also many other online streaming applications employing this model (Johannes et al., 2001). For example, in a telecom transaction stream, the scope of the stream is captured on a daily basic. Accordingly, a set of landmarks is given to the mining system and the results are computed from the current point back to the immediately-preceding landmark. We discuss some typical algorithms in this mining model.

Lossy Counting (Gurmeet and Rajeev, 2002) is one of the first algorithms to find all frequent sets from entire transactional streams. It is a deterministic approach since the mining results are guaranteed within some error. Given an error threshold ε and a minimum support threshold s, Lossy Counting guarantees that all itemsets whose true frequency exceeds sN are included in the mining results (where N is the number of transactions seen so far in the stream). Furthermore, the estimated frequency for each output itemset is guaranteed to be no more than its true frequency by an amount of εN. In Lossy Counting, the stream is divided into buckets, each has the same size of $w = 1/\varepsilon$ transactions and labeled with bucket ids, starting from 1. At transaction N, the bucket id is $b = \lceil N/w \rceil$. During the mining process, Lossy Counting maintains a synopsis S which is a set of entries (X, f, Δ), where X is an itemset, f is its estimated frequency, and Δ is the maximal possible error in f. For each X found in the stream, its frequency will be incremented by 1 if it is found in S; otherwise a new entry is created with the value $(X, f, b - 1)$, where $b = \lceil N/w \rceil$. At each bucket boundary (i.e., $N \equiv 0 \bmod w$), S is pruned by deleting all itemsets that have $f + \Delta \leq b$. The frequent itemsets are those having $f \geq (s - \varepsilon)N$.

To see how Lossy Counting approximates frequency errors, it is important to note that the stream is divided into equally sized buckets, and each itemset X will have its frequency tracked only if it appears frequently enough; i.e., on average, at least once on every bucket sized $1/\varepsilon$. Accordingly, the deletion rule ensures that whenever $f \leq b - \Delta$, X will be deleted. Note that, $(b - \Delta)$ is the number of buckets since the last time X occurs frequently enough to be counted. This value is always no more than N/w which is the number of buckets so far in the stream. Consequently, the pruning condition always makes sure that $f \leq b - \Delta \leq N/w$ or $f \leq \varepsilon N$. This is also the maximal lost on counting frequency of every itemset.

It is important to note that when Lossy Counting is applied to mining frequent itemsets, it needs to mine the stream in batch mode in order to avoid explicitly enumerating all subsets of every transaction. Consequently, this approach may limit its applicability in some online streaming environments (Johannes et al., 2001). Recently, this drawback has been addressed by EStream algorithm (Xuan Hong et al., 2006). In this algorithm, the error on the mining results is strictly guaranteed within a given threshold whilst the data stream is processed online. The basic idea of EStream is still based on the bucketing technique. However, it has been observed that when the transaction flow is processed online and the downward closure property is employed to find frequent sets, a longer itemset is usually being delayed counting until all its subsets are found potentially frequent. Accordingly, there will be a bigger error margin on frequent sets of larger sizes. Therefore, in EStream, the error (also frequency) of each itemset is identified precisely based on their length. This algorithm has been theoretically proven and guarantees that: (i) there is no error on frequency counting of 1-itemsets; (ii) for 2-itemsets, the maximal error on frequency counting is no more than εN; (iii) and for itemsets of length $k > 2$, the error is no more than $2\varepsilon N$ (where k is the maximal length of frequent itemsets).

The advantage of a deterministic approach is that it is able to produce all truly frequent itemsets and limit the maximal error on frequency counting. However, in order to provide this guarantee, a large amount of

memory space is needed to maintain itemsets whose support is greater than ε (meanwhile we only need to find those whose support is greater than s). Furthermore, since all itemsets having frequency $f \geq (s - \varepsilon)N$ are reported as frequent, this might include some itemsets whose frequency is between $(s - \varepsilon)N \leq f \leq sN$, and are not truly frequent. This gives the motivation to develop methods based on a probabilistic approach and FDPM (Jeffrey et al., 2004) is one of such algorithms.

FDPM (Frequent Data stream Pattern Mining): In FDPM, the Chernoff bound has been applied to mine frequent sets from entire data streams. Generally, the Chernoff bound can give us certain probability guarantees on the estimation of statistics of the underlying data when given a certain number of observations (transactions) regarding the data. In FDPM, the Chernoff bound is used to reduce the error of mining results when more and more transactions have been processed. For this reason, ε is a running value instead of a parameter specified by users. Its value will approach 0 when the number of observed transactions is very large.

In frequent set mining problems, an itemset X appearing in a transaction t_i can be viewed as a Bernoulli trial and thus one can denote a random variable $A_i=1$ if X occurs in t_i and $A_i=0$ if not. Now considering a sequence of transactions $t_1, t_2,...,t_n$, as n independent Bernoulli trials such that $\Pr(A_i = 1) = p$ and $\Pr(A_i = 0) = 1 - p$ for a probability p. Then, let r be the number of times that $A_i=1$ in these n transactions, with its expectation being np. The Chernoff bound states that, for any $\gamma > 0$:

$$\Pr\{| r - np | \geq np\gamma\} \leq 2e^{-np\gamma^2/2}$$

By dividing the left hand size by n and replacing p by s, $p\gamma$ by ε, and r/n by $\bar{r}$:

$$\Pr\{| \bar{r} - s | \geq \varepsilon\} \leq 2e^{-n\varepsilon^2/(2s)}$$

Let the right hand size be equal to δ, then:

$$\varepsilon = \sqrt{\frac{2s\ln(2/\delta)}{n}}$$

Now, given a sequence of transactions arriving in the stream $t_1, t_2,...,t_n, t_{n+1},...,t_N$, and assume that n first transactions have been observed ($n < N$). For an itemset X, we call *sup(X,n)* its running support in n transactions, and *sup(X,N)* its true support in N transactions. By replacing $\bar{r}$ by *sup(X,n)* and s by *sup(X,N)*, then when n transactions have been processed, *sup(X,n)* is beyond $\pm\varepsilon$ of *sup(X,N)* with probability no more than δ. In FDPM design, this observation is used to prune itemsets that may not potentially be frequent. A set of entries P is used to maintain frequent itemsets found in the stream. In case where each transaction in the stream is only one item, P's size is bounded by at most $1/(s - \varepsilon)$ entries. However, when each transaction is a set of items, P's size is identified via experiments (due to the nature of the exponential explosion of itemsets). Whenever P is full, those itemsets whose frequency is smaller than $(s - \varepsilon)n$ will be deleted from P. Recall that, since ε is reduced with time, this deletion condition makes P always keep all potential frequent itemsets. The advantage of FDPM is that the memory space used to maintain potentially frequent item(set)s is much smaller than those in the above deterministic algorithms. FDPM prunes itemsets whose support is smaller than $(s - \varepsilon)$ while the deterministic algorithms prune those smaller than ε. However, FDPM's drawback is that it assumes data arriving in the stream are independent (in order to apply the Chernoff bound). In reality, the characteristics of data streams may often change with time and the data is highly possible to be dependent. In these cases, the quality of FDPM may not be able to be guaranteed. Furthermore, this algorithm is false-negative oriented due to the basic of a probabilistic approach.

Forgetful Window Model

Different from the landmark window model, where all transactions appearing in the stream are equally important, in the forgetful window model, the old transactions will have less effect on the mining results comparing to the new arrival ones in the stream. Therefore, in this model, a fading function is often used to reduce the weight (or importance) of transactions as time goes by. Also, note that when transactions are reduced their weight, the importance of the frequent itemsets in which they appear is also reduced.

FP-Streaming (Giannella et al., 2003) is a typical algorithm addressing this model. In this work, the stream is physically divided into batches and frequent itemsets are discovered separately from each other. Their frequencies are reduced gradually by using a fading factor $\Phi<1$. For instance, let X be an itemset that has been found frequent in a set of consecutive batches

$B_i,\ldots,B_j$, where i, j are batch indices and $i<j$, then its frequency will be faded gradually by the formula

$$\sum_{k=i}^{j} \Phi^{j-k} f_X(B_k).$$

Note that since Φ is always smaller than 1, the older the B_k, the smaller the Φ^{j-k} value. Furthermore, in FP-Streaming, the frequency of an itemset will be maintained for every batch as long as it is still frequent; meanwhile the number of batches is increased with time. Therefore, in order to address the memory space constraint, a logarithm tilted-time window is utilized for each itemset in FP-Stream. It stores the itemset's frequency with fine granularity for recent batches and coarser granularity for older ones. Consequently, this approach not only requires less memory space but also allows to find changes or trends in the stream (by comparing itemsets' frequencies between different periods of time frame). This algorithm is also a deterministic approach since the error on frequency counting is guaranteed within a given threshold.

Sliding Window Model

Compared to the two above models, this model further considers the elimination of transactions. When new transactions come, the oldest ones will be eliminated from the mining model. Accordingly, the mining results are only discovered within a fixed portion of the stream which is pre-specified by a number of transactions (count-based sliding window) or by a time period (time-based sliding window). The sliding window model is quite suitable for various online data stream applications such as stock markets, financial analysis or network monitoring where the most valuable information is that discovered from the recently arriving portion of the stream. In the following, we review two typical algorithms proposed in this mining model.

Time Sensitive Sliding window: this algorithm is proposed to mine frequent sets from a time-based sliding window. The window is defined as a set of time intervals during which the rate of arrival transaction flow can change with time. Accordingly, the number of transactions arriving in each time interval (thus the sliding window) may not be fixed. Potentially frequent itemsets are then uncovered from each time interval. Their local frequencies are stored in a counting frequency table. Unlike the methods presented above (e.g., Lossy Counting, EStream, FDPM) where the memory space is affected by the error threshold, this method directly fixes the memory space for the counting frequency table. Consequently, whenever this table is full, there is a need to adjust it in order to get space for newly generated itemsets. The adjustment can be achieved by merging two adjacent entries, which have frequency counting found in the same time interval, into one entry. This new entry may select either the smaller or the larger value (between two original entries) for its frequency counting. Due to this procedure, the mining results can only be either false-positive oriented or false-negative oriented. Furthermore, in this algorithm, the error on frequency counting of each output itemset cannot be precisely estimated and bounded.

FTP-DS (Frequent Temporal Patterns of Data Streams) is another typical algorithm to mine frequent sets from a time-based sliding window (Wei-Guang et al., 2003). However, in this approach, the stream is divided into several disjoint sub-streams where each one represents the transaction flow of one customer. Accordingly, an itemset's frequency is defined as the ratio between the number of customers that subscribe to it and the total number of customers in the sliding window. When the window slides, the frequency of each itemset is then tracked for each time period. In order to address the issue of memory space constraint, the least mean square (LMS) method is applied. Therefore, a linear function $f = a + b \times t$ is used as a summary form to represent each itemset's frequency history. Given time period t, the itemset's frequency in this period can be derived. The interesting point of this approach is that both coefficients a and b can be updated incrementally. For example, given $\{(t_s, f_s),\ldots,(t_c, f_c)\}$ to be a set of time period and frequency count pairs for an itemset X from t_s to t_c, according to the LMS method, the sum of squared error (SSE) needs to be minimized:

$$SSE = \sum e_i^2 = \sum (f_i - a - b \times t_i)^2$$

(where i =s,…,c)

By differentiating SSE with respect to a and b, the formulas to compute two coefficients are:

$$b = (\sum f_i t_i - q^{-1} \sum f_i \sum t_i) \times (\sum t_i^2 - q^{-1} (\sum w_i)^2)^{-1}$$

and

$$a = q^{-1}(\sum f_i - b \times \sum t_i),$$

where $q = t_c - t_s + 1$. Note that two summations $\sum f_i t_i, \sum f_{i,}$ are easily updated incrementally at each time period t_i; meanwhile $\sum t_i, (\sum t_i)^2$ can be simply interpolated from t_s and t_c. Therefore, only 3 numbers $(t_s, \sum f_i t_i, \sum f_{i,})$ need to be maintained for each itemset, making the memory space independent from the data stream size. However, it is still an open problem to provide an error bound on the mining results when the LMS method is used in this algorithm.

FUTURE TRENDS

From the above discussion, it can be seen that most algorithms reported so far only focus on the problem of maintaining frequent itemsets within limited memory space. Some of them can provide error guarantees on the mining results. However, they all ignore the fact that CPU is also a bounded resource. In reality, for many stream applications such as stock monitoring or telecom fraud detection, the data arrival rate is rapid and mostly unpredictable (due to the dynamic changing behavior of data sources). Meanwhile, the mining results of these streams usually require delivery under time constraints to enable optimal decisions. Therefore, it is necessary to address the problem of controlling CPU workload in the context of finding frequent sets from online streaming environments (Xuan Hong et al., 2007). A natural approach is to drop some fraction of data streams in order to reduce workload (Nesime et al., 2007). However, it is necessary to address the following questions: How to determine overload situations? How much load needs to be shed? How to approximate frequent sets under the introduction of load shedding? More theories and techniques of load shedding certainly should be developed in the near future.

CONCLUSION

Data streams disseminate data in the form of continuous, rapid, unbound and fast changing behavior. Accordingly, algorithms designed in stream environments usually face the challenge of supporting one-pass data processing whilst working under the limitation of system resources. This chapter discussed this challenge in the context of mining frequent sets from streaming data. Most typical approaches, together with their mining models, have been presented and analyzed. However, these works are still in an early stage of studying data streams. Many issues remain open and new techniques must be developed to address them. Furthermore, finding frequent sets is also the core step in many data mining tasks including association rule mining, correlation analysis, clustering, and classification. It is believed that, when more effective algorithms are developed to find frequent sets in stream environments, their applicability will play a key role in various real-time making decision systems.

REFERENCES

Brian, B., Shivnath, B., Mayur, D., Rajeev, M., & Jennifer, W. (2002). Models and issues in data stream systems. *PODS Conference*, 1-16.

Chih-Hsiang, L., Ding-Ying, C., Yi-Hung, W., & Arbee, C. (2005). Mining frequent itemsets from data streams with a time-sensitive sliding window. *SIAM Conference*, 68-79.

Giannella, C., Han, J., Pei, J., Yan, X., & Philip S. Y. (2003). Mining Frequent Patterns in Data Streams at Multiple Time Granularities. *Next Generation Data Mining.AAAI/MIT*.

Gurmeet, S. M., & Rajeev, M. (2002). Approximate frequency counts over data streams. *VLDB Conference*, 346-357.

Jeffrey, X. Y., Zhihong, C., Hongjun, L., & Aoying, Z. (2004). False positive or false negative: Mining frequent itemsets from high speed transactional data streams. *VLDB Conference*, 204-215.

Johannes, G., Flip, K., Divesh, S. (2001). On Computing Correlated Aggregates Over Continual Data Streams. *SIGMOD Conference*, 13-24.

Minos, G., Johannes G., & Rajeev, R. (2002). Querying and mining data streams: you only get one look a tutorial. *ACM SIGMOD Conference*, 635-635.

Nesime, T., Ugur Ç., Stanley B. Z. (2007). Staying FIT: Efficient Load Shedding Techniques for Distributed Stream Processing. *VLDB Conference*, 159-170.

Pedro, D., & Geoff, H. (2001). Catching up with the data: Research issues in mining data streams. *ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*.

Wei-Guang, T., Ming-Syan, C., & Philip, S. Y. (2003). A regression-based temporal pattern mining scheme for data streams. *VLDB Conference*, 93-104.

Xuan Hong, D., Wee-Keong, Ng, & Kok-Leong, O. (2006). EStream: Online Mining of Frequent Sets with Precise Error Guarantee. *The 8th International Conference Data Warehousing and Knowledge Discovery*, 312-321.

Xuan Hong, D., Wee-Keong, Ng, Kok-Leong, O., & Vincent C. S. L. (2007). Discovering Frequent Sets from Data Streams with CPU Constraint. *The 6th Australian Data Mining Conference*.

KEY TERMS

Bernoulli Trial: In probability and statistics, an experiment is called a Bernoulli trial if its outcome is random and can take one of two values "true" or "false".

Binomial Distribution: A discrete probability distribution of the number of "true" outcomes in a sequence of n independent true/false experiments (Bernoulli trials), each of which yields "true" with probability p.

Deterministic Algorithm: In stream mining for frequent sets, an algorithm is deterministic if it can provide the mining results with *a priori* error guarantees.

False-Negative Oriented Approach: In stream mining for frequent sets, if an algorithm misses some truly frequent itemsets in the final results then it is called a false-negative oriented approach.

False-Positive Oriented Approach: In stream mining for frequent sets, if an algorithm outputs some infrequent itemsets in the final results then it is called a false-positive oriented approach.

Immediate Subset: Given an itemset X of length m, itemset Y is called its immediate subset if $Y \subseteq X$ and Y has length m-1.

Load Shedding: The process of dropping excess load from a mining system in order to guarantee the timely delivery of mining results.

Fuzzy Methods in Data Mining

Eyke Hüllermeier
Philipps-Universität Marburg, Germany

INTRODUCTION

Tools and techniques that have been developed during the last 40 years in the field of fuzzy set theory (FST) have been applied quite successfully in a variety of application areas. A prominent example of the practical usefulness of corresponding techniques is *fuzzy control*, where the idea is to represent the input-output behaviour of a controller (of a technical system) in terms of fuzzy rules. A concrete control function is derived from such rules by means of suitable inference techniques.

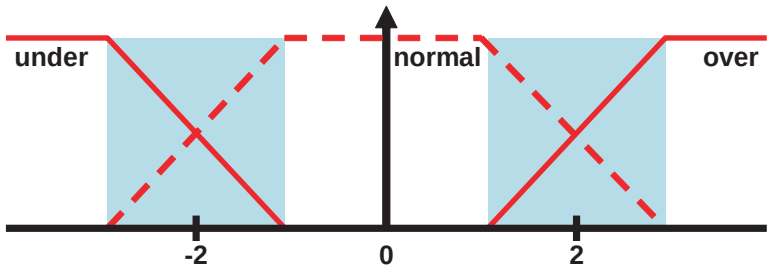
While aspects of knowledge representation and reasoning have dominated research in FST for a long time, problems of *automated learning and knowledge acquisition* have more and more come to the fore in recent years. There are several reasons for this development, notably the following: Firstly, there has been an internal shift within fuzzy systems research from "modelling" to "learning", which can be attributed to the awareness that the well-known "knowledge acquisition bottleneck" seems to remain one of the key problems in the design of intelligent and knowledge-based systems. Secondly, this trend has been further amplified by the great interest that the fields of *knowledge discovery in databases* (KDD) and its core methodical component, *data mining*, have attracted in recent years.

It is hence hardly surprising that data mining has received a great deal of attention in the FST community in recent years (Hüllermeier, 2005). The aim of this chapter is to give an idea of the usefulness of FST for data mining. To this end, we shall briefly highlight, in the next but one section, some potential advantages of fuzzy approaches. In preparation, the next section briefly recalls some basic ideas and concepts from FST. The style of presentation is purely non-technical throughout; for technical details we shall give pointers to the literature.

BACKGROUND ON FUZZY SETS

A fuzzy subset F of a reference set X is identified by a so-called *membership function* (often denoted $\mu_F(\bullet)$), which is a generalization of the characteristic function of an ordinary set $A \subseteq X$ (Zadeh, 1965). For each element $x \in X$, this function specifies the *degree of membership* of x in the fuzzy set. Usually, membership degrees $\mu_F(x)$ are taken from the unit interval $[0,1]$, i.e., a membership function is an $X \rightarrow [0,1]$ mapping, even though more general membership scales (such as ordinal scales or complete lattices) are conceivable.

Figure 1. Fuzzy partition of the gene expression level with a "smooth" transition (grey regions) between underexpression, normal expression, and overexpression



Fuzzy sets formalize the idea of *graded membership* according to which an element can belong "more or less" to a set. Consequently, a fuzzy set can have "non-sharp" boundaries. Many sets or concepts associated with natural language terms have boundaries that are non-sharp in the sense of FST. Consider the concept of "forest" as an example. For many collections of trees and plants it will be quite difficult to decide in an unequivocal way whether or not one should call them a forest.

In a data mining context, the idea of "non-sharp" boundaries is especially useful for discretizing numerical attributes, a common preprocessing step in data analysis. For example, in gene expression analysis, one typically distinguishes between *normally expressed*, *underexpressed*, and *overexpressed* genes. This classification is made on the basis of the expression level of the gene (a normalized numerical value), as measured by so-called DNA-chips, by using corresponding thresholds. For example, a gene is often called overexpressed if its expression level is at least twofold increased. Needless to say, corresponding thresholds (such as 2) are more or less arbitrary. Figure 1 shows a *fuzzy partition* of the expression level with a "smooth" transition between under-, normal, and overexpression. For instance, according to this formalization, a gene with an expression level of at least 3 is definitely considered overexpressed, below 1 it is definitely not overexpressed, but in-between, it is considered overexpressed to a certain degree (Ortoloani et al., 2004).

Fuzzy sets or, more specifically, membership degrees can have different semantic interpretations. Particularly, a fuzzy set can express three types of cognitive concepts which are of major importance in artificial intelligence, namely *uncertainty*, *similarity*, and *preference* (Dubois, 1997). To operate with fuzzy sets in a formal way, FST offers generalized set-theoretical respectively logical connectives and operators (as in the classical case, there is a close correspondence between set-theory and logic) such as triangular norms (t-norms, generalized logical conjunctions), t-conorms (generalized disjunctions), and generalized implication operators. For example, a t-norm $\otimes$ is a $[0,1]\times[0,1]\rightarrow[0,1]$ mapping which is associative, commutative, monotone increasing (in both arguments) and which satisfies the boundary conditions $\alpha \otimes 0 = 0$ and $\alpha \otimes 1 = \alpha$ for all $0 \leq \alpha \leq 1$ (Klement et al., 2002). Well-known examples of t-norms include the minimum $(\alpha, \beta) \mapsto \min(\alpha, \beta)$ and the product $(\alpha, \beta) \mapsto \alpha\beta$.

BENEFITS OF FUZZY DATA MINING

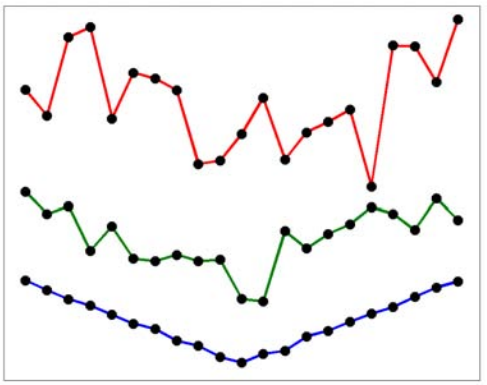
This section briefly highlights some potential contributions that FST can make to data mining; see (Hüllermeier, 2005) for a more detailed (technical) exposition.

Graduality

The ability to represent gradual concepts and fuzzy properties in a thorough way is one of the key features of fuzzy sets. This aspect is also of primary importance in the context of data mining. In fact, patterns that are of interest in data mining are often inherently vague and do have boundaries that are non-sharp in the sense of FST. To illustrate, consider the concept of a "peak": It is usually not possible to decide in an unequivocal way whether a timely ordered sequence of measurements, such as the expression profile of a gene, has a "peak" (a particular kind of pattern) or not. Rather, there is a gradual transition between having a peak and not having a peak. Likewise, the spatial extension of patterns like a "cluster of points" or a "region of high density" in a data space will usually have soft rather than sharp boundaries.

Many data mining methods proceed from a representation of the entities under consideration in terms of *feature vectors*, i.e., a fixed number of features or attributes, each of which represents a certain property

Figure 2. Three exemplary time series that are more or less "decreasing at the beginning"



of an entity (e.g., the age of a person). Having defined a suitable feature set, a common goal is to analyze relationships and dependencies between the attributes. In this regard, the possibility to model graded properties in an adequate way is useful for both feature extraction and subsequent dependency analysis.

To illustrate, consider again a (discrete) time series of the form $x = (x(1), x(2)...x(n))$, e.g., a gene expression profile where $x(t)$ is the expression level at time point t. For a profile of that kind, it might be of interest whether or not it is "decreasing at the beginning". This property is inherently fuzzy: First, it is unclear which time points belong to the "beginning", and defining these points in a non-fuzzy way by a "crisp" subset $B = \{1,2...k\}$ comes along with a certain arbitrariness (the choice of threshold $k \in \{1...n\}$) and does not appear fully convincing. Second, the intended meaning of "decreasing at the beginning" will not be captured well by the standard mathematical definition, which requires

$$\forall t \in B : x(t) \geq x(t+1), \quad (1)$$

In fact, the human perception of "decreasing" will usually be tolerant toward small violations of Eqn. 1, especially if such violations may be caused by noise in the data; see Figure 2 for an illustration, where the second profile is still considered as "decreasing at the beginning", at least to some extent. FST offers a large repertoire of techniques and concepts for generalizing the description of a property at a formal (logical) level, including generalized logical connectives such as the aforementioned t-norms and t-conorms, fuzzy GREATER-THAN relations (which generalize the above $\geq$ relation), and fuzzy FOR-MOST quantifiers (which generalize the universal quantifier in Eqn. 1). This way, it becomes possible to specify the degree to which a profile is "decreasing at the beginning", i.e., to characterize a profile in terms of a fuzzy feature which assumes a value in the unit interval [0,1] instead of a binary feature which is either present (1) or absent (0). See (Lee et al., 2006) for an interesting application, where corresponding modeling techniques have been used in order to formalize so-called candlestick patterns which are of interest in stock market analysis.

The increased expressiveness offered by fuzzy methods is also advantageous for the modeling of patterns in the form of relationships and dependencies between different features. As a prominent example of a (local) pattern, consider the concept of an association rule $A \rightarrow B$ suggesting that, if the antecedent A holds, then typically the consequent B holds as well (Agrawal & Srikant, 1994). In the non-fuzzy case, a rule is either satisfied (supported) by a concrete example (data entity) or not. In the fuzzy case, a rule can again be satisfied (supported) to a certain degree (Chen et al., 2003). At a formal level, the modeling of a rule makes use of generalized logical operators such as t-norms (for combining the conditions in the antecedent part) and implication operators (for combining the antecedent and the consequent part). Depending on the concrete operators employed, a fuzzy rule can have different semantic interpretations. For example, the meaning of gradual THE MORE—THE MORE dependencies such as "the more a profile is decreasing at the beginning, the more it increases at the end" can be captured in an adequate way by means of so called residuated implication operators; we refer to (Dubois et al., 2006) for a technical discussion of these issues.

In a data mining context, taking graduality into account can be very important in order to decide whether a certain pattern is "interesting" or not. For example, one important criterion in this regard is *frequency*, i.e., the number of occurrences of a pattern in a data set. If a pattern is specified in an overly restrictive (non-fuzzy) manner, it can easily happen that none of the entities matches the specification, even though many of them can be seen as approximate matches. In such cases, the pattern might still be considered as "well-supported" by the data (Dubois et al., 2005).

Regarding computational aspects, one may wonder whether the problem to extract interesting fuzzy patterns from large data sets, i.e., patterns that fulfil a number of generalized (fuzzy) criteria, can still be solved by efficient algorithms that guarantee scalability. Fortunately, efficient algorithmic solutions can be assured in most cases, mainly because fuzzy extensions can usually resort to the same algorithmic principles as non-fuzzy methods. Consequently, standard algorithms can most often be used in a modified form, even though there are of course exceptions to this rule.

Linguistic Representation and Interpretability

A primary motivation for the development of fuzzy sets was to provide an interface between a numerical scale and a symbolic scale which is usually composed

of linguistic terms. Thus, fuzzy sets have the capability to interface quantitative patterns with qualitative knowledge structures expressed in terms of natural language. This makes the application of fuzzy technology very appealing from a knowledge representational point of view, as it allows patterns discovered in a database to be presented in a linguistic and hence comprehensible way. For example, given a meaningful formalization of the concepts "multilinguality" and "high income" in terms of fuzzy sets, it becomes possible to discover an association rule in an employee database which expresses a dependency between these properties and can be presented to a user in the form "multilinguality usually implies high income" (Kacprzyk and Zadrozny, 2005).

Without any doubt, the user-friendly representation of models and patterns is often rightly emphasized as one of the key features of fuzzy methods (Mencar et al., 2005). Still, this potential advantage should be considered with caution in the context of data mining. A main problem in this regard concerns the high subjectivity and context-dependency of fuzzy patterns: A rule such as "multilinguality usually implies high income" may have different meanings to different users of a data mining system, depending on the concrete interpretation of the fuzzy concepts involved (multilinguality, high income). It is of course possible to disambiguate a model by complementing it with the semantics of these concepts (including the specification of membership functions). Then, however, the complete model, consisting of a qualitative (linguistic) and a quantitative part, becomes more cumbersome.

To summarize on this score, the close connection between a numerical and a linguistic level for representing patterns, as established by fuzzy sets, can indeed help a lot to improve interpretability of patterns, though linguistic representations also involve some complications and should therefore not be considered as preferable per se.

Robustness

Robustness is often emphasized as another advantage of fuzzy methods. In a data mining context, the term "robustness" can of course refer to many things. Generally, a data mining method is considered robust if a small variation of the observed data does hardly alter the induced model or the evaluation of a pattern. Another desirable form of robustness of a data mining method is robustness toward variations of its parameterization: Changing the parameters (e.g., the interval-boundaries of a histogram, Viertl & Trutschnig, 2006) slightly should not have a dramatic effect on the output of the method.

Even though the topic of robustness is still lacking a thorough theoretical foundation in the fuzzy set literature, it is true that fuzzy methods can avoid some obvious drawbacks of conventional interval-based approaches for dealing with numerical attributes. For example, the latter can produce quite undesirable "threshold effects": A slight variation of an interval boundary can have a very strong effect on the evaluation of patterns such as association rules (Sudkamp, 2005). Consequently, the set of "interesting" patterns can be very sensitive toward the discretization of numerical attributes.

Representation of Uncertainty

Data mining is inseparably connected with uncertainty (Vazirgiannis et al., 2003). For example, the data to be analyzed is imprecise, incomplete, or noisy most of the time, a problem that can badly deteriorate a mining algorithm and lead to unwarranted or questionable results. But even if observations are perfect, the alleged "discoveries" made in that data are of course afflicted with uncertainty. In fact, this point is especially relevant for data mining, where the systematic search for interesting patterns comes along with the (statistical) problem of multiple hypothesis testing, and therefore with a high danger of making false discoveries.

Fuzzy sets and the related uncertainty calculus of *possibility theory* have made important contributions to the representation and processing of uncertainty. In data mining, like in other fields, related uncertainty formalisms can complement probability theory in a reasonable way, because not all types of uncertainty relevant to data mining are of a probabilistic nature, and because other formalisms are in some situations more expressive than probability. For example, probability is not very suitable for representing ignorance, which might be useful for modeling incomplete or missing data.

FUTURE TRENDS

Looking at the current research trends in data mining, which go beyond analyzing simple, homogeneous, and

static data sets, fuzzy methods are likely to become even more important in the near future. For example, fuzzy set-based modeling techniques may provide a unifying framework for modeling complex, uncertain, and heterogeneous data originating from different sources, such as text, hypertext, and multimedia data. Moreover, fuzzy methods have already proved to be especially useful for mining data and maintaining patterns in dynamic environments where patterns "smoothly" evolve in the course of time (Beringer & Hüllermeier, 2007).

Finally, FST seems to be especially qualified for data pre- and post-processing, e.g., for data summarization and reduction, approximation of complex models and patterns in terms of "information granules" (Hüllermeier, 2007), or the (linguistic) presentation of data mining results. Even though current research is still more focused on the mining process itself, this research direction is likely to become increasingly important against the background of the aforementioned trend to analyze complex and heterogeneous information sources.

CONCLUSIONS

Many features and patterns of interest in data mining are inherently fuzzy, and modeling them in a non-fuzzy way will inevitably produce results which are unsatisfactory in one way or the other. For example, corresponding patterns may not appear meaningful from a semantic point of view or lack transparency. Moreover, non-fuzzy methods are often not very robust and tend to overlook patterns which are only approximately preserved in the data. Fuzzy methods can help to alleviate these problems, especially due to their increased expressiveness that allows one to represent features and patterns in a more adequate and flexible way.

REFERENCES

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules. In *Proceedings of the 20th Conference on VLDB*, pages 487-499. Santiago, Chile.

Beringer, J., & Hüllermeier, E. (2007). Fuzzy Clustering of Parallel Data Streams. In Valente de Oliveira J., & Pedrycz W. (Eds.), *Advances in Fuzzy Clustering and Its Application*. John Wiley and Sons, 2007.

Chen,G., Wei, Q., Kerre, E., & Wets, G. (2003). Overview of fuzzy associations mining. In *Proc. ISIS-2003, 4th International Symposium on Advanced Intelligent Systems*. Jeju, Korea.

Dubois, D., Hüllermeier, E., & Prade, H. (2006). A systematic approach to the assessment of fuzzy association rules. *Data Mining and Knowledge Discovery*, 13(2):167-192.

Dubois D., & Prade, H. (1997). The three semantics of fuzzy sets. *Fuzzy Sets and Systems*, 90(2):141-150.

Dubois, D., Prade, H., & Sudkamp T. (2005). On the representation, measurement, and discovery of fuzzy associations. *IEEE Transactions on Fuzzy Systems*, 13(2):250-262.

Hüllermeier, E. (2005). Fuzzy sets in machine learning and data mining: Status and prospects. *Fuzzy Sets and Systems*, 156(3):387-406.

Hüllermeier, E. (2007). Granular Computing in Machine Learning and Data Mining. In Pedrycz, W., Skowron, A., & Kreinovich, V. (Eds.). *Hanbook on Granular Computing*. John Wiley and Sons.

Kacprzyk, J., & Zadrozny, S. (2005). Linguistic database summaries and their protoforms: towards natural language based knowledge discovery tools. *Information Sciences*, 173(4):281-304.

Klement, EP., Mesiar, R., & Pap, E. (2002). *Triangular Norms*. Dordrecht: Kluwer Academic Publishers.

Lee, CHL., Liu, A., & Chen, WS. (2006). Pattern discovery of fuzzy time series for financial prediction. *IEEE Transactions on Knowledge and Data Engineering*, 18(5):613-625.

Mencar C., Castellano, G., & Fanelli, A.M. (2005). Some Fundamental Interpretability Issues in Fuzzy Modeling. *Proceedings of EUSFLAT-2005*, Valencia, Spain.

Ortolani, M., Callan, O., Patterson, D, & Berthold, M. (2004). Fuzzy subgroup mining for gene associations. *Processing NAFIPS-2004*, pages 560-565, Banff, Alberta, Canada.

Sudkamp, T. (2005). Examples, counterexamples, and measuring fuzzy associations. *Fuzzy Sets and Systems*, 149(1).

Vazirgiannis, M., Halkidi, M., & Gunopoulos, D. (2003). Quality Assessment and Uncertainty Handling in Data Mining. Berlin: Springer-Verlag. 2003.

Viertl, R. & Trutschnig, W. (2006) *Fuzzy histograms and fuzzy probability distributions. Proceedings IPMU-2006*, Paris: Editions EDK.

Zadeh, L.A. (1965). Fuzzy sets. *Information and Control*, 8(3):338-353.

KEY TERMS

Fuzzy Feature: A fuzzy feature is a property of an object (data entity) that can be present to a certain degree. Formally, a fuzzy feature is modelled in terms of a fuzzy set and can be considered as a generalization of a binary feature.

Fuzzy Partition: A fuzzy partition of a set X is a (finite) collection of fuzzy subsets which cover X in the sense that every $x \in X$ belongs to at least one fuzzy set with a non-zero degree of membership. Fuzzy partitions are often used instead of interval-partitions in order to discretize the domain of numerical attributes.

Fuzzy Pattern: A fuzzy pattern is a pattern the representation of which involves concepts from fuzzy set theory, i.e., membership functions or generalized logical operators. A typical example is a fuzzy association rule, which connects two fuzzy features by means of a generalized logical connective such as an implication operator.

Fuzzy Set: A fuzzy set is a set that allows for a partial belonging of elements. Formally, a fuzzy set of a given reference X set is defined by a membership function that assigns a membership degree (from an ordered scale, usually the unit interval) to every element of X.

Generalized Logical Operators: Generalizations of Boolean connectives (negation, conjunction, disjunction, implication) which operate on a fuzzy membership scale (usually the unit interval). In particular, the class of so-called triangular norms (with associated triangular co-norms) generalizes the logical conjunction (disjunction).

Interpretability: In a data mining context, interpretability refers to the request that the results produced by a mining algorithm should be transparent and understandable by a (human) user of the data mining system. Amongst others, this requires a low complexity of models and patterns shown to the user and a representation in terms of cognitively comprehensible concepts.

Robustness: In a data mining context, robustness of methods and algorithms is a desirable property that can refer to different aspects. Most importantly, a method should be robust in the sense that the results it produces are not excessively sensitive toward slight variations of its parameterization or small changes of the (input) data.

A General Model for Data Warehouses

Michel Schneider
Blaise Pascal University, France

INTRODUCTION

Basically, the schema of a data warehouse lies on two kinds of elements: facts and dimensions. Facts are used to memorize measures about situations or events. Dimensions are used to analyse these measures, particularly through aggregation operations (counting, summation, average, etc.). To fix the ideas let us consider the analysis of the sales in a shop according to the product type and to the month in the year. Each sale of a product is a fact. One can characterize it by a quantity. One can calculate an aggregation function on the quantities of several facts. For example, one can make the sum of quantities sold for the product type "mineral water" during January in 2001, 2002 and 2003. Product type is a criterion of the dimension Product. Month and Year are criteria of the dimension Time. A quantity is so connected both with a type of product and with a month of one year. This type of connection concerns the organization of facts with regard to dimensions. On the other hand a month is connected to one year. This type of connection concerns the organization of criteria within a dimension. The possibilities of fact analysis depend on these two forms of connection and on the schema of the warehouse. This schema is chosen by the designer in accordance with the users needs.

Determining the schema of a data warehouse cannot be achieved without adequate modelling of dimensions and facts. In this article we present a general model for dimensions and facts and their relationships. This model will facilitate greatly the choice of the schema and its manipulation by the users.

BACKGROUND

Concerning the modelling of dimensions, the objective is to find an organization which corresponds to the analysis operations and which provides strict control over the aggregation operations. In particular it is important to avoid double-counting or summation of non-additive data. Many studies have been devoted to this problem. Most recommend organizing the criteria (we said also members) of a given dimension into hierarchies with which the aggregation paths can be explicitly defined. In (Pourabbas, 1999), hierarchies are defined by means of a containment function. In (Lehner, 1998), the organization of a dimension results from the functional dependences which exist between its members, and a multi-dimensional normal form is defined. In (Hùsemann, 2000), the functional dependences are also used to design the dimensions and to relate facts to dimensions. In (Abello, 2001), relationships between levels in a hierarchy are apprehended through the Part-Whole semantics. In (Tsois, 2001), dimensions are organized around the notion of a dimension path which is a set of drilling relationships. The model is centered on a parent-child (one to many) relationship type. A drilling relationship describes how the members of a children level can be grouped into sets that correspond to members of the parent level. In (Vassiliadis, 2000), a dimension is viewed as a lattice and two functions "anc" and "desc" are used to perform the roll up and the drill down operations. Pedersen (1999) proposes an extended multidimensional data model which is also based on a lattice structure, and which provides non-strict hierarchies (i.e. too many relationships between the different levels in a dimension).

Modelling of facts and their relationships has not received so much attention. Facts are generally considered in a simple fashion which consists in relating a fact with the roots of the dimensions. However, there is a need for considering more sophisticated structures where the same set of dimensions are connected to different fact types and where several fact types are inter-connected. The model described in (Pedersen, 1999) permits some possibilities in this direction but is not able to represent all the situations.

Apart from these studies it is important to note various propositions (Agrawal, 1997; Datta, 1999; Gyssens, 1997; Nguyen, 2000) for cubic models where the primary objective is the definition of an algebra for multidimensional analysis. Other works must also be mentioned. In (Golfarelli, 1998), a solution is proposed to

derive multidimensional structures from E/R shemas. In (Hurtado, 2001) are established conditions for reasoning about summarizability in multidimensional structures.

MAIN THRUST

Our objective in this article is to propose a generic model based on our personal research work and which integrates existing models. This model can be used to apprehend the sharing of dimensions in various ways and to describe different relationships between fact types. Using this model, we will also define the notion of well-formed warehouse structures. Such structures have desirable properties for applications. We suggest a graph representation for such structures which can help the users in designing and requesting a data warehouse.

Modelling Facts

A fact is used to record measures or states concerning an event or a situation. Measures and states can be analysed through different criteria organized in dimensions.

A fact type is a structure

fact_name[(fact_key),
(list_of_reference_attributes), (list_of_fact_attributes)]

where

- fact_name is the name of the type;
- fact_key is a list of attribute names; the concatenation of these attributes identifies each instance of the type;
- list_of_reference_attributes is a list of attribute names; each attribute references a member in a dimension or another fact instance;
- list_of_fact_attributes is a list of attribute names; each attribute is a measure for the fact.

The set of referenced dimensions comprises the dimensions which are directly referenced through the list_of_reference_attributes, but also the dimensions which are indirectly referenced through other facts.

Each fact attribute can be analysed along each of the referenced dimensions. Analysis is achieved through the computing of aggregate functions on the values of this attribute.

As an example let us consider the following fact type for memorizing the sales in a set of stores.

Sales[(ticket_number, product_key), (time_key, product_key, store_key),
(price_per_unit, quantity)]

The key is (ticket_number, product_key). This means that there is an instance of Sales for each different product of a ticket. There are three dimension references: time_key, product_key, store_key. There are two fact attributes: price_per_unit, quantity. The fact attributes can be analysed through aggregate operations by using the three dimensions.

There may be no fact attribute; in this case a fact records the occurrence of an event or a situation. In such cases, analysis consists in counting occurrences satisfying a certain number of conditions.

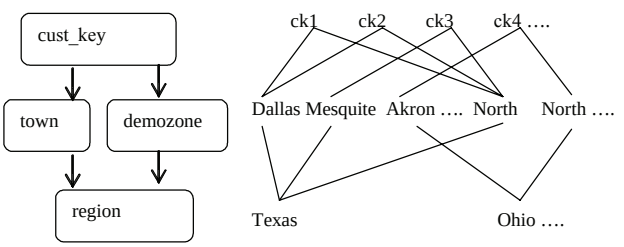
For the needs of an application, it is possible to introduce different fact types sharing certain dimensions and having references between them.

Modelling Dimensions

The different criteria which are needed to conduct analysis along a dimension are introduced through members. A member is a specific attribute (or a group of attributes) taking its values on a well defined domain. For example, the dimension TIME can include members such as DAY, MONTH, YEAR, etc. Analysing a fact attribute A along a member M means that we are interested in computing aggregate functions on the values of A for any grouping defined by the values of M. In the article we will also use the notation Mij for the j-th member of i-th dimension.

Members of a dimension are generally organized

Figure 1. A typical hierarchy in a dimension



in a hierarchy which is a conceptual representation of the hierarchies of their occurrences. Hierarchy in dimensions is a very useful concept that can be used to impose constraints on member values and to guide the analysis. Hierarchies of occurrences result from various relationships which can exist in the real world: categorization, membership of a subset, mereology. Figure 1 illustrates a typical situation which can occur. Note that a hierarchy is not necessarily a tree.

We will model a dimension according to a hierarchical relationship (HR) which links a child member Mij (i.e. town) to a parent member Mik (i.e. region) and we will use the notation Mij–Mik. For the following we consider only situations where a child occurrence is linked to a unique parent occurrence in a type. However, a child occurrence, as in case (b) or (c), can have several parent occurrences but each of different types. We will also suppose that HR is reflexive, antisymmetric and transitive. This kind of relationship covers the great majority of real situations. Existence of this HR is very important since it means that the members of a dimension can be organized into levels and correct aggregation of fact attribute values along levels can be guaranteed.

Each member of a dimension can be an entry for this dimension i.e. can be referenced from a fact type. This possibility is very important since it means that dimensions between several fact types can be shared in various ways. In particular, it is possible to reference a dimension at different levels of granularity. A dimension root represents a standard entry. For the three dimensions in Figure 1, there is a single root. However, definition 3 authorizes several roots.

As in other models (Hùsemann, 2000), we consider property attributes which are used to describe the members. A property attribute is linked to its member through a functional dependence, but does not introduce a new member and a new level of aggregation. For example the member *town* in the *customer* dimension may have property attributes such as population, administrative position, etc. Such attributes can be used in the selection predicates of requests to filter certain groups.

We now define the notion of member type, which incorporates the different features presented above.

A member type is a structure:

member_name[(member_key),
dimension_name,
(list_of_reference_attributes)]

where

- member_name is the name of the type;
- member_key is a list of attribute names; the concatenation of these attributes identifies each instance of the type;
- list_of_reference_attributes is a list of attribute names where each attribute is a reference to the successors of the member instance in the cover graph of the dimension.

Only the member_key is mandatory.

Using this model, the representations of the members of dimension *customer* in Figure 1 are the following:

cust_root[(cust_key), customer, (town_name, zone_name)]
town[(town_name), customer,(region_name)]
demozone[(zone_name), customer, (region_name)]
region[(region_name), customer, ()]

Well-Formed Structures

In this section we explain how the fact types and the member types can be interconnected in order to model various warehouse structures.

First, a fact can directly reference any member of a dimension. Usually a dimension is referenced through one of its roots (as we saw above, a dimension can have several roots). But it is also interesting and useful to have references to members other than the roots. This means that a dimension can be used by different facts with different granularities. For example, a fact can directly reference *town* in the *customer* dimension and another can directly reference *region* in the same

Figure 2. Typical warehouse structures

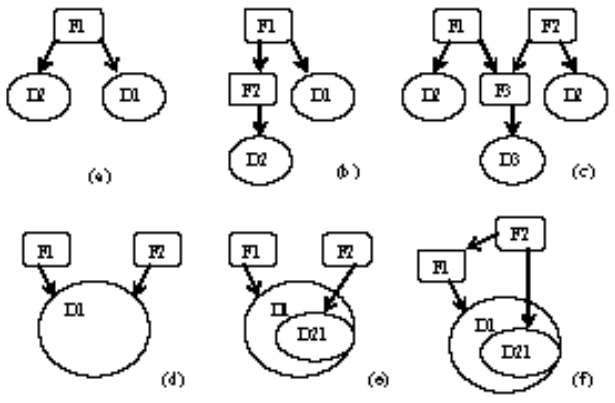
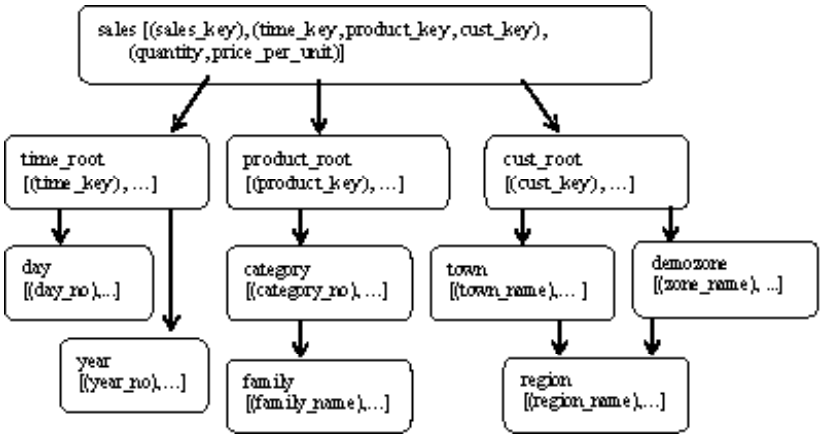


Figure 3. The DWG of a star-snowflake structure



dimension. This second reference corresponds to a coarser granule of analysis than the first.

Moreover, a fact F1 can reference any other fact F2. This type of reference is necessary to model real situations. This means that a fact attribute of F1 can be analysed by using the key of F2 (acting as the grouping attribute of a normal member) and also by using the dimensions referenced by F2.

To formalise the interconnection between facts and dimensions, we thus suggest extending the HR relationship of section 3 to the representation of the associations between fact types and the associations between fact types and member types. We impose the same properties (reflexivity, anti-symmetry, transitivity). We also forbid cyclic interconnections. This gives a very uniform model since fact types and member types are considered equally. To maintain a traditional vision of the data warehouses, we also ensure that the members of a dimension cannot reference facts.

Figure 2 illustrates the typical structures we want to model. Case (a) corresponds to the simple case, also known as star structure, where there is a unique fact type F1 and several separate dimensions D1, D2, etc. Cases (b) and (c) correspond to the notion of facts of fact. Cases (d), (e) and (f) correspond to the sharing of the same dimension. In case (f) there can be two different paths starting at F2 and reaching the same member M of the sub-dimension D21. So analysis using these two paths cannot give the same results when reaching M. To pose this problem we introduce the DWG and the path coherence constraint.

To represent data warehouse structures, we suggest using a graph representation called DWG (data warehouse graph). It consists in representing each type (fact type or member type) by a node containing the main information about this type, and representing each reference by a directed edge.

Suppose that in the DWG graph, there are two different paths P1 and P2 starting from the same fact type F, and reaching the same member type M. We can analyse instances of F by using P1 or P2. The path coherence constraint is satisfied if we obtain the same results when reaching M.

For example in case of figure 1 this constraint means the following: for a given occurrence of cust_key, whether the town path or the demozone path is used, one always obtains the same occurrence of region.

We are now able to introduce the notion of well-formed structures.

Definition of a Well-Formed Warehouse Structure

A warehouse structure is well-formed when the DWG is acyclic and the path coherence constraint is satisfied for any couple of paths having the same starting node and the same ending node.

A well-formed warehouse structure can thus have several roots. The different paths from the roots can be always divided into two sub-paths: the first one with only fact nodes and the second one with only member nodes. So, roots are fact types.

Since the DWG is acyclic, it is possible to distribute its nodes into levels. Each level represents a level of aggregation. Each time we follow a directed edge, the level increases (by one or more depending on the used path). When using aggregate operations, this action corresponds to a ROLLUP operation (corresponding to the semantics of the HR) and the opposite operation to a DRILLDOWN. Starting from the reference to a dimension D in a fact type F, we can then roll up in the hierarchies of dimensions by following a path of the DWG.

Illustrating the Modelling of a Typical Case with Well-Formed Structures

Well-formed structures are able to model correctly and completely the different cases of Figure 2. We illustrate in this section the modelling for the star-snowflake structure.

We have a star-snowflake structure when:

- there is a unique root (which corresponds to the unique fact type);
- each reference in the root points towards a separate subgraph in the DWG (this subgraph corresponds to a dimension).

Our model does not differentiate star structures from snowflake structures. The difference will appear with the mapping towards an operational model (relational model for example). The DWG of a star-snowflake structure is represented in Figure 3. This representation is well-formed. Such a representation can be very useful to a user for formulating requests.

FUTURE TRENDS

A first opened problem is that concerning the property of summarizability between the levels of the different dimensions. For example, the total of the sales of a product for 2001 must be equal to the sum of the totals for the sales of this product for all months of 2001. Any model of data warehouse has to respect this property. In our presentation we supposed that function HR verified this property. In practice, various functions were proposed and used. It would be interesting and useful to begin a general formalization which would regroup all these propositions.

Another opened problem concerns the elaboration of a design method for the schema of a data warehouse. A data warehouse is a data base and one can think that its design does not differ from that of a data base. In fact a data warehouse presents specificities which it is necessary to take into account, notably the data loading and the performance optimization. Data loading can be complex since sources schemas can differ from the data warehouse schema. Performance optimization arises particularly when using relational DBMS for implementing the data warehouse.

CONCLUSION

In this paper we propose a model which can describe various data warehouse structures. It integrates and extends existing models for sharing dimensions and for representing relationships between facts. It allows for different entries in a dimension corresponding to different granularities. A dimension can also have several roots corresponding to different views and uses. Thanks to this model, we have also suggested the concept of Data Warehouse Graph (DWG) to represent a data warehouse schema. Using the DWG, we define the notion of well-formed warehouse structures which guarantees desirable properties.

We have illustrated how typical structures such as star-snowflake structures can be advantageously represented with this model. Other useful structures like those depicted in Figure 2 can also be represented.

The DWG gathers the main information from the warehouse and it can be very useful to users for formulating requests. We believe that the DWG can be used as an efficient support for a graphical interface to manipulate multidimensional structures through a graphical language.

The schema of a data warehouse represented with our model can be easily mapped into an operational model. Since our model is object-oriented a mapping towards an object model is straightforward. But it is possible also to map the schema towards a relational model or an object relational model. It appears that our model has a natural place between the conceptual schema of the application and an object relational implementation of the warehouse. It can thus serve as a helping support for the design of relational data warehouses.

REFERENCES

Abello, A., Samos, J., & Saltor, F. (2001). Understanding analysis dimensions in a multidimensional object-oriented model. *Intl Workshop on Design and Management of Data Warehouses, DMDW'2000,* Interlaken, Switzerland.

Agrawal, R., Gupta, A., & Sarawagi, S. (1997). Modelling multidimensional databases. *International Conference on Data Engineering, ICDE'97* (pp. 232-243), Birmingham, UK.

Datta, A., & Thomas, H. (1999). The cube data model: A conceptual model and algebra for on-line analytical processing in data warehouses. *Decision Support Systems, 27(*3), 289-301.

Golfarelli, M., Maio, D., & Rizzi, V.S. (1998). Conceptual design of data warehouses from E/R schemes. *32th Hawaii International Conference on System Sciences, HICSS'1998*.

Gyssens, M., & Lakshmanan, V.S. (1997). A foundation for multi-dimensional databases. *Intl Conference on Very Large Databases* (pp. 106-115).

Hurtado, C., & Mendelzon, A. (2001). Reasoning about summarizability in heterogeneous multidimensional schemas. *International Conference on Database Theory, ICDT'01*.

Hùsemann, B., Lechtenbörger, J., & Vossen, G. (2000). Conceptual data warehouse design. *Intl Workshop on Design and Management of Data Warehouse, DMDW'2000,* Stockholm, Sweden.

Lehner, W., Albrecht, J., & Wedekind, H. (1998). Multidimensional normal forms. *l0th Intl Conference on Scientific and Statistical Data Management, SSDBM'98,* Capri, Italy.

Nguyen, T., Tjoa, A.M., Wanger, S. (2000). Conceptual multidimensional data model based on metacube. *Intl Conference on Advances in Information Systems* (pp. 24-33), Izmir, Turkey.

Pedersen, T.B., & Jensen, C.S. (1999). Multidimensional data modelling for complex data. *Intl Conference on Data Engineering, ICDE'99*.

Pourabbas, E., & Rafanelli, M. (1999). Characterization of hierarchies and some operators in OLAP environment. *ACM Second International Workshop on Data Warehousing and OLAP, DOLAP'99* (pp. 54-59), Kansas City, USA.

Tsois, A., Karayannidis, N., & Sellis, T. (2001). MAC: Conceptual data modeling for OLAP. *Intl Workshop on Design and Management of Data Warehouses, DMDW'2000,* Interlaken, Switzerland.

Vassiliadis, P., & Skiadopoulos, S. (2000). Modelling and optimisation issues for multidimensional databases. *International Conference on Advanced Information Systems Engineering, CAISE'2000* (pp. 482-497), Stockholm, Sweden.

KEY TERMS

Data Warehouse: A data base which is specifically elaborated to allow different analysis on data. Analysis consists generally to make aggregation operations (count, sum, average, etc.). A data warehouse is different from a transactional data base since it accumulates data along time and other dimensions. Data of a warehouse are loaded and updated at regular intervals from the transactional data bases of the company.

Dimension: Set of members (criteria) allowing to drive the analysis (example for the Product dimension: product type, manufacturer type). Members are used to drive the aggregation operations.

Drilldown: Opposite operation of the previous one.

Fact: Element recorded in a warehouse (example: each product sold in a shop) and whose characteristics (i.e. measures) are the object of the analysis (example: quantity of a product sold in a shop).

Galaxy Structure: Structure of a warehouse for which two different types of facts share a same dimension.

Hierarchy: The members of a dimension are generally organized along levels into a hierarchy.

Member: Every criterion in a dimension is materialized through a member.

Rollup: Operation consisting in going in a hierarchy at a more aggregated level.

Star Structure: Structure of a warehouse for which a fact is directly connected to several dimensions and can be so analyzed according to these dimensions. It is the most simple and the most used structure.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 523-528, copyright 2005 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

A Genetic Algorithm for Selecting Horizontal Fragments

Ladjel Bellatreche
Poitiers University, France

INTRODUCTION

Decision support applications require complex queries, e.g., multi way joins defining on huge warehouses usually modelled using star schemas, i.e., a fact table and a set of data dimensions (Papadomanolakis & Ailamaki, 2004). Star schemas have an important property in terms of join operations between dimensions tables and the fact table (i.e., the fact table contains foreign keys for each dimension). None join operations between dimension tables. Joins in data warehouses (called star join queries) are particularly expensive because the fact table (the largest table in the warehouse by far) participates in every join and multiple dimensions are likely to participate in each join.

To speed up star join queries, many optimization structures were proposed: redundant structures (materialized views and advanced index schemes) and non redundant structures (data partitioning and parallel processing). Recently, data partitioning is known as an important aspect of physical database design (Sanjay, Narasayya & Yang, 2004; Papadomanolakis & Ailamaki, 2004). Two types of data partitioning are available (Özsu & Valduriez, 1999): vertical and horizontal partitioning. Vertical partitioning allows tables to be decomposed into disjoint sets of columns. Horizontal partitioning allows tables, materialized views and indexes to be partitioned into disjoint sets of rows that are physically stored and usually accessed separately. Contrary to redundant structures, data partitioning does not replicate data, thereby reducing storage requirement and minimizing maintenance overhead. In this paper, we concentrate only on horizontal data partitioning (HP).

HP may affect positively (1) query performance, by performing *partition elimination*: if a query includes a partition key as a predicate in the WHERE clause, the query optimizer will automatically route the query to only relevant partitions and (2) database manageability: for instance, by allocating partitions in different machines or by splitting any access paths: tables, materialized views, indexes, etc. Most of database systems allow three methods to perform the HP using PARTITION statement: RANGE, HASH and LIST (Sanjay, Narasayya & Yang, 2004). In the range partitioning, an access path (table, view, and index) is split according to a range of values of a given set of columns. The hash mode decomposes the data according to a hash function (provided by the system) applied to the values of the partitioning columns. The list partitioning splits a table according to the listed values of a column. These methods can be combined to generate composite partitioning. Oracle currently supports range-hash and range-list composite partitioning using PARTITION - SUBPARTITION statement. The following SQL statement shows an example of fragmenting a table Student using range partitioning.

CREATE TABLE student (Student_ID Number(6), Student_FName VARCHAR(25), Student-LName VARCHAR(25), PRIMARY KEY (Student_ID) PARTITION BY *RANGE* (student_LN) (PARTITION student_ae VALUES LESS THAN ('F%') TABLESPACE part1, PARTITION student_fl VALUES LESS THAN ('M%') TABLESPACE part2, PARTITION student_mr VALUES LESS THAN ('S%') TABLESPACE part3, PARTITION student_sz VALUES LESS THAN (MAXVALUE) TABLESPACE part4);

HP can also be combined with others optimization structures like indexes, materialized views, and parallel processing. Several work and commercial systems show its utility and impact in optimizing OLAP queries (Noaman & Barker, 1999, Sanjay, Narasayya & Yang, 2004; Stöhr, Märtens & Rahm, 2000). But none of these studies has formalized the problem of selecting a horizontal partitioning schema to speed up a set of queries as optimization problem with constraint and proposed selection algorithms.

Logically, two types of HP are distinguished (Ceri, Negri & Pelagatti, 1982; Özsu & Valduriez, 1999): (1) *primary HP* and (2) *derived HP*. The primary HP consists in fragmenting a table T based only on its attribute(s). The derived HP consists in splitting a table S (e.g., the fact table of a given star schema) based on fragmentation schemas of other tables (e.g., dimension tables). This type has been used in (Stöhr, Märtens & Rahm, 2000). The primary HP may be used in optimizing selection operations, while the second in optimizing join and selection operations since it pre-computes them.

BACKGROUND

The HP in relational data warehouses is more challenging compared to that in relational and object databases. Several work and commercial systems show its utility and impact in optimizing OLAP queries (Sanjay, Narasayya & Yang, 2004; Stöhr, Märtens & Rahm,2000). But none study has formalized the problem of selecting a horizontal partitioning schema to speed up a set of queries as an optimization problem and proposed selection algorithms.

This challenge is due to the several choices of partitioning schema (a fragmentation schema is the result of the data partitioning process) of a star or snowflake schemas:

1. Partition only the dimension tables using simple predicates defined on these tables (a simple predicate p is defined by: p: A_i θ Value; where A_i is an attribute, θ ∈{=, <, >, ≥, ≤}, and Value ∈ Domain(A_i)). This scenario is not suitable for OLAP queries, because the sizes of dimension tables are generally small compare to the fact table. Most of OLAP queries access the fact table, which is huge Therefore, any partitioning that does not take into account the fact table is discarded.
2. Partition only the fact table using simple predicates defined on this table because it normally contains millions of rows and is highly normalized. The fact relation stores time-series factual data. The fact table is composed of foreign keys and raw data. Each foreign key references a primary key on one of the dimension relations. These dimension relations could be time, product, customer, etc. The raw data represent the numerical measurement of the organization such as sales amount, number of units sold, prices and so forth. The raw data usually never contain descriptive (textual) attributes because the fact relation is designed to perform arithmetic operations such as summarization, aggregation, average and so forth on such data. In a data warehouse modelled by a star schema, most of OLAP queries access dimension tables first and after that to the fact table. This choice is also discarded.
3. Partition some/all dimension tables using their predicates, and then partition the fact table based on the fragmentation schemas of dimension tables. This approach is best in applying partitioning in data warehouses. Because it takes into consideration the star join queries requirements and the relationship between the fact table and dimension tables. In our study, we recommend the use of this scenario.

Horizontal Partitioning Selection Problem

Suppose a relational warehouse modelled by a star schema with d dimension tables and a fact table F. Among these dimension tables, we consider that g tables are fragmented (g ≤ d), where each table D_i (1 ≤ i ≤ g) is partitioned into m_i fragments: {D_{i1}, D_{i2}, …, D_{imi}}, such as: $D_{ij} = \sigma_{cl_{ji}}(D_i)$, where cl_{ji} and σ represent a conjunction of simple predicates and the selection predicate, respectively. Thus, the fragmentation schema of the fact table F is defined as follows:

$F_i = F \lceil D_{1i} \lceil D_{2i} \lceil \ldots \lceil D_{gi}$, $(1 \le i \le m_i)$, where $\lceil$ represents the semi join operation.

Example

To illustrate the procedure of fragmenting the fact table based on the fragmentation schemas of dimension tables, let's consider a star schema with three dimension tables: Customer, Time and Product and one fact table Sales. Suppose that the dimension table Customer is fragmented into two fragments *CustFemale* and *CustMale* defined by the following clauses:

CustFemale = $\sigma_{(Sex = `F')}$ (Customer) and CustMale = $\sigma_{(Sex = `M')}$ (Customer).

Therefore, the fact table *Sales* is fragmented based on the partitioning of Customer into two fragments $Sales_1$ and $Sales_2$ such as: $Sales_1$ = Sales $\ltimes$ CustFemale and $Sales_2$ = Sales $\ltimes$ CustMale. The initial star schema (Sales, Customer, Product, Time) is represented as the juxtaposition of two sub star schemas S_1 and S_2 such as: S_1: ($Sales_1$, CustFemale, Product, Time) (sales activities for only female customers) and S_2: ($Sales_2$, CustMale, Product, Time) (sales activities for only male customers). Note that the attributes used for partitioning are called fragmentation attributes (Sex attribute).

The number of horizontal fragments of the fact table generated by the partitioning procedure is given by the following equation:

$$N = \prod_{i=1}^{g} m_i ,$$

where m_i represents the number of fragments of the dimension table D_i. This number may be very large. For example, suppose we have: Customer dimension table partitioned into 50 fragments using the State attribute (case of 50 states in the U.S.A.), Time into 36 fragments using the Month attribute (if the sale analysis is done based on the three last years), and Product into 80 fragments using Package_type attribute, therefore the fact table will be fragmented into **144 000** fragments (50 * 36 * 80). Consequently, instead to manage one star schema, the data warehouse administrator (DWA) will manage 144 000 sub star schemas. It will be very hard for him to maintain all these sub-star schemas.

A formulation of the horizontal partitioning problem is the following:

Given a set of dimension tables D = {D_1, D_2, ..., D_d} and a set of OLAP queries Q = {Q_1, Q_2, ..., Q_m}, where each query Q_i ($1 \leq i \leq m$) has an access frequency. The HP problem consists in determining a set of dimension tables $D' \subseteq D$ to be partitioned and generating a set of fragments that can be used to partition the fact table F into a set of horizontal fragments (called fact fragments) {F_1, F_2, ..., F_N} such that: **(1)** The sum of the query cost when executed on top of the partitioned star schema is minimized; and **(2)** $N \leq W$, where W is a threshold fixed by the DWA representing the maximal number of fragments that he can maintain. This threshold is chosen based on the nature of the most frequently queries that contain selection conditions. Each The respect of this constraint avoids an explosion of the number of the fact fragments.

To deal with the HP selection problem, we use a genetic algorithm (Bellatreche, Boukhalfa & Abdalla, 2006).

MAIN FOCUS

We use a genetic algorithm (GA) to select a HP schema. The motivation to use GAs in solving horizontal partitioning problem was based on the observations that data warehouse can have a large number of fragmentation schemas. GAs has been largely used in databases. For instance, in optimizing query joins (Ioannidis & Kang, 1990) and selecting materialized views (**Zhang & Yang,**1999). GAs (Holland, 1975) are search methods based on the evolutionary concept of natural mutation and the survival of the fittest individuals. Given a well-defined search space they apply three different genetic search operations, namely, selection, crossover, and mutation, to transform an initial population of chromosomes, with the objective to improve their quality.

Fundamental to the GA structure is the notion of chromosome, which is an encoded representation of a feasible solution. Before the search process starts, a set of chromosomes is initialized to form the first generation. Then the three genetic search operations are repeatedly applied, in order to obtain a population with better characteristics. An outline of a GA is as follows: (see Box 1.).

Different steps of the GA: encoding mechanism, selection, crossover and mutation operators are presented in the following sections.

The most difficult part of applying GA is the representation of the solutions that represent fragmentation schemas. Traditionally, GAs used binary strings

Box 1.

```
Generate initial population;
Perform selection step;
While (stopping criterion not met) do
        Perform Crossover step;
        Perform mutation step;
        Perform Selection step;
End while
Report the best chromosome as the final solution.
```

to represent their chromosomes. In this study, another representation is proposed.

Coding Mechanism

Note that any horizontal fragment is defined by a clause of conjunctive predicates (minterm) defined on fragmentation attributes (Bellatreche & Boukhalfa, 2005). Before presenting our coding mechanism of dimension fragment, we should identify which dimension tables will participate in fragmenting the fact table. To do so, we propose the following procedure: (1) extract all simple predicates used by the m queries, (2) assign to each dimension table D_i ($1 \leq i \leq d$) its set of simple predicates, denoted by $SSPD_i$, (3) dimension tables having an empty SSPD will not be considered in the partitioning process.

Note that each fragmentation attribute has a domain of values. The clauses of horizontal fragments partition each attribute domain into sub domains. Consider two fragmentation attributes[1] Age and Gender of dimension table CUSTOMER and one attribute Season of dimension table TIME. The domains of these attributes are defined as follows: Dom(Age) =]0, 120], Dom(Gender) = {'M', 'F'}, and Dom(Season) = {"Summer", "Spring", "Autumn", "Winter"}. Suppose that on the attribute Age, three simple predicates are defined as follows: P_1 : (Age $\leq$ 18), P_2 : (Age $\geq$ 60), et P_3 : (18 < Age < 60). The domain of this attribute is then partitioned into three sub domains (see Figure 1 (a)): Dom(Age) = $d_{11} \cup d_{12} \cup d_{13}$, with d_{11} =]0, 18], d_{12} =]18, 60[, d_{13} = [60, 120]. Similarly, the domain of Gender attribute is decomposed into two sub domains: Dom(Gender) = $d_{21} \cup d_{22}$, with d_{21} = {'M'}, d_{22} = {'F'}. Finally, domain of Season is partitioned into four sub domains: Dom(Season) = $d_{31} \cup d_{32} \cup d_{33} \cup d_{34}$, where d_{31} = {"Summer"}, d_{32} = {"Spring"}, d_{33} = {"Autumn"}, and d_{34} = {"Winter"}.

The decomposition of each fragmentation attribute domain can be represented by an array with n_i cells, where n_i corresponds to number of its sub domains. The values of these cells are between 1 and n_i. Consequently, each chromosome (fragmentation schema) can be represented by a multi-dimensional array. From this representation, the obtainment of horizontal fragments is straightforward. It consists in generating all conjunctive clauses. If two cells of the same array have a same value, then they will be merged (or-ed).

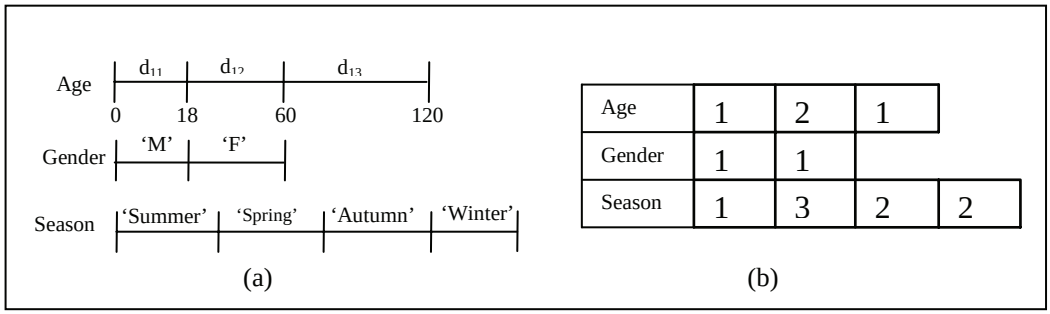
For example, in Figure 1(b), we can deduce that the fragmentation of the data warehouse is not performed using the attribute Gender, because all its sub domains have the same value. Consequently, the warehouse will be fragmented using only Season and Age. For Season attribute, three simple predicates are possible: P_1 : Season = "Summer", P_2 : Season = "Spring", and P_3 : (Season = "Autumn") OR (Season = "Winter"). For Age attribute, two predicates are possible: P_4 : (Age $\leq$ 18) OR (Age $\geq$ 60) and P_5 : (18 < Age < 60) Therefore, the data warehouse can be fragmented into six fragments defined by the following clauses: Cl_1 : ($P_1 \wedge P_4$); Cl_2 : ($P_1 \wedge P_5$); Cl_3 : ($P_2 \wedge P_4$); Cl_4 : ($P_2 \wedge P_5$); Cl_5 : ($P_3 \wedge P_4$); and Cl_6 : ($P_3 \wedge P_5$).

The coding that we proposed satisfies the correctness rules (completeness, reconstruction and disjointness (Özsu & Valduriez, 1999). It is used to represent fragments of dimension tables and fact table. This multidimensional array representation may be used to generate all possible fragmentation schemas (an exhaustive search). This number is calculated as:

$$2^{\sum_{i=1}^{K} n_i},$$

where K represents the number of fragmentation attributes. This number is equal to the number of minterms

Figure 1. Sub Domains of fragmentation attributes



generated by the horizontal fragmentation algorithm proposed by (Özsu & Valduriez, 1999).

The selection operation of GAs gives the probability of chromosomes being selected for reproduction. The principle is to assign higher probabilities to filter chromosomes. The roulette wheel method is used in our algorithm. The roulette wheel selection scheme allocates a sector on a roulette wheel to each chromosome. The ratio of angles of two adjacent sectors is a constant. The basic idea of rank based roulette wheel selection is to allocate larger sector angle to better chromosomes so that better solutions will be included in the next generation with higher probability. A new chromosome for the next generation is cloned after a chromosome as an offspring if a randomly generated number falls in the sector corresponding to the chromosome. Alternatively, the proportionate selection scheme generates chromosomes for the next generation only from a predefined percentage of the previous population.

The goodness of each each chromosome is evaluated using the cost model defined in (Bellatreche, Boukhalfa & Abdalla, 2006).

We use a two-point crossover mechanism for the following reason (note that fragments are represented by arrays): The chromosomes are crossed over once for each fragmentation attribute. If the crossover is done over one chromosome, an attribute with high number of sub domains (example of attribute Season that has 4 sub domains) will have a probability greater than attribute with low number of sub domains, like gender (with two sub domains). This operation is applied until none reduction of the number fact fragments fixed by the DBA.

The mutation operation is needed to create new chromosomes that may not be present in any member of a population and enables GA to explore all possible solutions (in theory) in the search space. The experimental details of our GA are given in (Bellatreche, Boukhalfa & Abdalla, 2006).

FUTURE TRENDS

Selecting horizontal data partitioning schema will become a challenging problem in physical database design. Other types of algorithms, like simulated annealing, ant colony, etc. may be used. Since in the data warehouse environment, changes are very frequently, it will be interesting to develop dynamic algorithms that take into account different changes (query changes, access frequency of a query, selectivity factor of predicates, size of tables, etc.). The horizontal partitioning can be also combined with other optimization structures like materialized views and indexes.

CONCLUSION

Data warehouse is usually modelled using relational database schema with huge fact table and dimension tables. On the top of this schema, complex queries are executed. To speed up these queries and facilitate the manageability of the huge warehouse, we propose the use of horizontal partitioning. It consists in decomposing different tables of the warehouse into fragments. First, we propose a methodology for fragmenting the fact table based on the fragmentation schemas of dimension tables. Second we formalize the problem of selecting an optimal horizontal partitioning schema as an optimization problem with constraint representing the number of horizontal fragments that the data warehouse administrator can manage. Finally, we propose a genetic algorithm to deal with problem.

REFERENCES

Bäck. T. (1995). Evolutionary algorithms in theory and practice. Oxford University Press, New York.

Bellatreche L. & Boukhalfa K. (2005). An Evolutionary Approach to Schema Partitioning Selection in a Data Warehouse, 7th International Conference on Data Warehousing and Knowledge Discovery, 115-125

Bellatreche L., Boukhalfa K. & Abdalla H. I. (2006). SAGA : A Combination of Genetic and Simulated Annealing Algorithms for Physical Data Warehouse Design, *in 23rd British National Conference on Databases*.

Bennett. K. P., Ferris M. C. & Ioannidis Y. E. (1991). A genetic algorithm for database query optimization. in Proceedings of the 4th International Conference on Genetic Algorithms, 400–407.

Ceri S., Negri M. & Pelagatti G. (1982). Horizontal data partitioning in database design, Proceedings of the ACM SIGMOD International Conference on Management of Data, 128-136.

Holland J. H. (1975). Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, Michigan.

Ioannidis. Y. & Kang Y. (1990). Randomized algorithms for optimizing large join queries. Proceedings of the ACM International Conference on Management of Data (SIGMOD), 9-22.

Özsu. M. T. & Valduriez P. (1999). Principles of Distributed Database Systems: Second Edition. Prentice Hall.

Noaman, A. Y. & Barker, K. (1999). A Horizontal Fragmentation Algorithm for the Fact Relation in a Distributed Data Warehouse. ACM CIKM, 154-161

Sanjay A., Narasayya V. R. & Yang B. (2004). Integrating vertical and horizontal partitioning into automated physical database design. Proceedings of the ACM International Conference on Management of Data (SIGMOD), 359–370.

Stöhr T., Märtens H. & Rahm E.(2000). Multi-dimensional database allocation for parallel data warehouses. Proceedings of the International Conference on Very Large Databases, 273–284.

Papadomanolakis, S. & Ailamaki A.(2004). Autopart: Automating schema design for large scientific databases using data partitioning. Proceedings of the 16th International Conference on Scientific and Statistical Database Management, 383–392.

Zhang C. & Yang J. (1999) Genetic Algorithm for Materialized View Selection in Data Warehouse Environments. Proceedings of the First International Conference on Data Warehousing and Knowledge Discovery, 116 – 125.

KEY TERMS

Data Partitioning: Distributing the rows of a table into several separate fragments.

Dimension Table: A table containing the data for one dimension within a star schema. The primary key is used to link to the fact table, and each level in the dimension has a corresponding field in the dimension table.

Derived Horizontal Fragmentation: Is the partitioning of a relation that results from simple predicates being defined on another relation.

Fact Table: The central table in a star schema, containing the basic facts or measures of interest. Dimension fields are also included (as foreign keys) to link to each dimension table.

Fitness Function: Is defined over the genetic representation and measures the quality of the represented solution.

Genetic Algorithm: Is search methods based on the evolutionary concept of natural mutation and the survival of the fittest individuals.

Minterm Predicate: Is the conjunction of simple predicates.

Primary Horizontal Partitioning: Primary horizontal fragmentation of a table is performed using predicates of queries accessing this table.

Selection Predicate: Selection predicate is the parameter of the selection operator of the relational algebra. It has the following form: Attribute θ Value; where Attribute is an attribute of a given table, $\theta \in \{=, <, >, \geq, \leq\}$, and Value $\in$ Domain(A_i)).

Selectivity of a Simple Predicate: Estimates the number of tuples from a relation that satisfy the simple predicate within a query.

Genetic Programming

William H. Hsu
Kansas State University, USA

INTRODUCTION

Genetic programming (**GP**) is a sub-area of evolutionary computation first explored by John Koza (1992) and independently developed by Nichael Lynn Cramer (1985). It is a method for producing computer programs through adaptation according to a user-defined fitness criterion, or objective function.

Like genetic algorithms, GP uses a representation related to some computational model, but in GP, fitness is tied to task performance by specific program semantics. Instead of strings or permutations, genetic programs are most commonly represented as variable-sized expression trees in imperative or functional programming languages, as grammars (O'Neill & Ryan, 2001), or as circuits (Koza et al., 1999). GP uses patterns from biological evolution to evolve programs:

- **Crossover:** Exchange of genetic material such as program subtrees or grammatical rules
- **Selection:** The application of the fitness criterion to choose which individuals from a population will go on to reproduce
- **Replication:** The propagation of individuals from one generation to the next
- **Mutation:** The structural modification of individuals

To work effectively, GP requires an appropriate set of program operators, variables, and constants. Fitness in GP is typically evaluated over fitness cases. In data mining, this usually means training and validation data, but cases can also be generated dynamically using a simulator or directly sampled from a real-world problem solving environment. GP uses evaluation over these cases to measure performance over the required task, according to the given fitness criterion.

BACKGROUND

Although Cramer (1985) first described the use of crossover, selection, and mutation and tree representations for using genetic algorithms to generate programs, Koza is indisputably the field's most prolific and persuasive author. (Wikipedia, 2007) In four books since 1992, Koza et al. have described GP-based solutions to numerous toy problems and several important real-world problems.

> **State of the field:** To date, GPs have been successfully applied to a few significant problems in machine learning and data mining, most notably symbolic regression and feature construction. The method is very computationally intensive, however, and it is still an open question in current research whether simpler methods can be used instead. These include supervised inductive learning, deterministic optimization, randomized approximation using non-evolutionary algorithms (such as Markov chain Monte Carlo approaches), or genetic algorithms and evolutionary algorithms. It is postulated by GP researchers that the adaptability of GPs to structural, functional, and structure-generating solutions of unknown form makes them more amenable to solving complex problems. Specifically, Koza et al. demonstrate (1999, 2003) that in many domains, GP is capable of "human-competitive" automated discovery of concepts deemed to be innovative through technical review such as patent evaluation.

MAIN THRUST OF THE CHAPTER

The general strengths of genetic programs lie in their ability to produce solutions of variable functional form, reuse partial solutions, solve multi-criterion optimization problems, and explore a large search space of solutions in parallel. Modern GP systems are also able to produce structured, object-oriented, and functional

programming solutions involving recursion or iteration, subtyping, and higher-order functions.

A more specific advantage of GPs are their ability to represent procedural, generative solutions to pattern recognition and machine learning problems. Examples of this include image compression and reconstruction (Koza, 1992) and several of the recent applications surveyed below.

GP Methodology

GP in pattern classification departs from traditional supervised inductive learning in that it evolves solutions whose functional form is not determined in advance, and in some cases can be theoretically arbitrary. Koza (1992, 1994) developed GPs for several pattern reproduction problems such as the multiplexer and symbolic regression problems.

Since then, there has been continuing work on inductive GP for pattern classification (Kishore et al., 2000), prediction (Brameier & Banzhaf, 2001), and numerical curve-fitting (Nikolaev & Iba, 2001, IEEE Trans. Evol. Comp.). GP has been used to boost performance in learning polynomial functions (Nikolaev & Iba, 2001, GP & Evol. Machines). More recent work on tree-based multi-crossover schemes has produced positive results in GP-based design of classification functions (Muni et al., 2004).

While early work in GP for data mining and knowledge discovery in databases (KDD) focused on specific fitness measures related to classification and prediction (Eggermont *et al.*, 1999), more recent work has sought to use GP to implement search behaviors and procedural solutions. Among the methodologies related to GP are swarm intelligence approaches such as ant colony optimization (ACO) and particle swarm optimization (PSO), which seek to evolve solutions through fine-grained simulation of many simple agents. (Azzag *et al.*, 2007; Holden & Freitas, 2007; Tsang & Kwong, 2007)

Applications in Data Mining and Warehousing

The domains within data mining and warehousing where GP has been most successfully applied in recent research include classification (Raymer et al., 1996; Connolly, 2004a; Langdon & Buxton, 2004; Langdon & Barrett, 2005; Holden & Freitas, 2007), prediction (Kaboudan, 2000), and search (Burke & Kendall, 2005). Higher-level tasks such as decision support are often reduced to classification or prediction, while the symbolic representation (S-expressions) used by GP admits query optimization.

GP for Control of Inductive Bias, Feature Construction, and Feature Extraction

GP approaches to inductive learning face the general problem of optimizing inductive bias: the preference for groups of hypotheses over others on bases other than pure consistency with training data or other fitness cases. Krawiec (2002) approaches this problem by using GP to preserve useful components of representation (features) during an evolutionary run, validating them using the classification data, and reusing them in subsequent generations. This technique is related to the wrapper approach to KDD, where validation data is held out and used to select examples for supervised learning, or to construct or select variables given as input to the learning system. Because GP is a generative problem solving approach, feature construction in GP tends to involve production of new variable definitions rather than merely selecting a subset.

Evolving dimensionally-correct equations on the basis of data is another area where GP has been applied. Keijzer & Babovic (2002) provide a study of how GP formulates its declarative bias and preferential (search-based) bias. In this and related work, it is shown that proper units of measurement (strong typing) approach can capture declarative bias towards correct equations, whereas type coercion can implement even better preferential bias.

Grammar-Based GP for Data Mining

Not all GP-based approaches use expression tree-based representations, nor functional program interpretation as the computational model. Wong and Leung (2000) survey data mining using grammars and formal languages. This general approach has been shown effective for some natural language learning problems, and extension of the approach to procedural informa-

tion extraction is a topic of current research in the GP community.

GP Implementations: Functionality and Research Features

A number of GP software packages are publicly and commercially available. General features common to most GP systems for research and development include: a very high-period random number generator such as the Mersenne Twister for random constant generation and GP operations; a variety of selection, crossover, and mutation operations; and trivial parallelism (e.g., through multithreading).

One of the most popular packages for experimentation with GP is *Evolutionary Computation in Java*, or *ECJ* (Luke *et al.*, 2007). *ECJ* implements the above features as well as parsimony, "strongly-typed" GP, migration strategies for exchanging individual subpopulations in island mode or multi-deme GP, vector representations, and reconfigurabity using parameter files.

In addition to standalone software packages, some systems used for industral-grade applications provide an interface between a GP module and a relational database server. These include SQL (Connolly, 2004a; Ishida & Pozo, 2002) and Microsoft form-based (Connolly, 2004b) interfaces.

Other Applications: Optimization, Policy Learning

Like other genetic and evolutionary computation methodologies, GP is driven by fitness and suited to optimization approaches to machine learning and data mining. Its program-based representation makes it good for acquiring policies by reinforcement learning. Many GP problems are "error-driven" or "payoff-driven" (Koza, 1992), including the ant trail problems and foraging problems now explored more heavily by the swarm intelligence and ant colony optimization communities. A few problems use specific information-theoretic criteria such as maximum entropy or sequence randomization.

FUTURE TRENDS

Limitations: Scalability and Solution Comprehensibility

Genetic programming remains a controversial approach due to its high computational cost, scalability issues, and current gaps in fundamental theory for relating its performance to traditional search methods, such as hill climbing. While GP has achieved results in design, optimization, and intelligent control that are as good as and sometimes better than those produced by human engineers, it is not yet widely used as a technique due to these limitations in theory. An additional controversy in the intelligent systems community is the role of knowledge in search-driven approaches such as GP. Some proponents of GP view it as a way to generate innovative solutions with little or no domain knowledge, while critics have expressed skepticism over original results due to the lower human-comprehensibility of some results. The crux of this debate is a tradeoff between innovation and originality versus comprehensibility, robustness, and ease of validation. Successes in replicating previously-patented engineering designs such as analog circuits using GP (Koza et al., 2003) have increased its credibility in this regard.

Open Issues: Code Growth, Diversity, Reuse, and Incremental Learning

Some of the most important open problems in GP deal with the proliferation of solution code (called code growth or code bloat), the reuse of previously-evolved partial solutions, and incremental learning. Code growth is an increase in solution size across generations, and generally refers to one that is not matched by a proportionate increase in fitness. It has been studied extensively in the field of GP by many researchers. Luke (2000) provides a survey of known and hypothesized causes of code growth, along with methods for monitoring and controlling growth. Recently Burke et al. (2004) explored the relationship between diversity (variation among solutions) and code growth and fitness. Some techniques for controlling

code growth include reuse of partial solutions through such mechanisms as automatically-defined functions, or ADFs (Koza, 1994) and incremental learning – that is, learning in stages. One incremental approach in GP is to specify criteria for a simplified problem and then transfer the solutions to a new GP population (Hsu & Gustafson, 2002).

CONCLUSION

Genetic programming (GP) is a search methodology that provides a flexible and complete mechanism for machine learning, automated discovery, and cost-driven optimization. It has been shown to work well in many optimization and policy learning problems, but scaling GP up to most real-world data mining domains is a challenge due to its high computational complexity. More often, GP is used to evolve data transformations by constructing features, or to control the declarative and preferential inductive bias of the machine learning component. Making GP practical poses several key questions dealing with how to scale up; make solutions comprehensible to humans and statistically validate them; control the growth of solutions; reuse partial solutions efficiently; and learn incrementally.

Looking ahead to future opportunities and challenges in data mining, genetic programming provides one of the more general frameworks for machine learning and adaptive problem solving. In data mining, they are likely to be most useful where a generative or procedural solution is desired, or where the exact functional form of the solution – whether a mathematical formula, grammar, or circuit – is not known in advance.

REFERENCES

Azzag, H., Guinot, C., & Venturini, G. (2007). Data and text mining with hierarchical clustering ants. In Grosan, C., Abraham, A., & Chis, M.,(Eds): *Swarm Intelligence in Data Mining*, pp. 153-189. Berlin, Germany: Springer.

Burke, E. K. & Kendall, G. (2005). *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques.* London, UK: Springer.

Burke, E. K., Gustafson, S. & Kendall, G. (2004). Diversity in genetic programming: an analysis of measures and correlation with fitness. *IEEE Transactions on Evolutionary Computation* **8**(1), p. 47-62.

Connolly, B. (2004a). SQL, Data Mining, & Genetic Programming. *Dr. Dobb's Journal*, April 4, 2004. Retrieved from: http://www.ddj.com/184405616

Connolly, B. (2004b). Survival of the fittest: natural selection with windows Forms. In John J. (Ed.), *MSDN Magazine, August 4, 2004*. Retrieved from: http://msdn.microsoft.com/msdnmag/issues/04/08/GeneticAlgorithms/

Cramer, Nichael Lynn (1985), A representation for the adaptive generation of simple sequential programs in: *Proceedings of the International Conference on Genetic Algorithms and their Applications (ICGA)*, Grefenstette, Carnegie Mellon University.

Eggermont, J., Eiben, A. E., & van Hemert, J. I. (1999). Adapting the fitness function in GP for data mining. In Poli, R., Nordin, P., Langdon, W. B., & Fogarty, T. C., (Eds), In *Proceedings of the Second European Workshop on Genetic Programming* (EuroGP 1999), Springer LNCS 1598, (pp. 195-204). Göteborg, Sweden, May 26-27, 1999.

Holden, N. & Freitas, A. A. (2007). A hybrid PSO/ACO algorithm for classification. In *Proceedings of the Genetic and Evolutionary Computation Conference* (GECCO 2007), (pp. 2745 - 2750). London, UK, July 7-11, 2007.

Hsu, W. H. & Gustafson. S. M. (2002). Genetic Programming and Multi-Agent Layered Learning by Reinforcements. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2002)*, New York, NY.

Ishida, C. Y. & Pozo, A. T. R. (2002). GPSQL Miner: SQL-grammar genetic programming in data mining. In *Proceedings of the 2002 Congress on Evolutionary Computation (CEC 2002)*, (pp. 1226-1231). Honolulu, USA, May 12-17, 2002.

Kaboudan, M. A. (2000). Genetic programming prediction of stock prices. *Computational Economics*, *16*(3):207-236.

Keijzer, M. & Babovic, V. (2002). Declarative and preferential bias in gp-based scientific discovery. *Genetic Programming and Evolvable Machines 3*(1), 41-79.

Kishore, J. K., Patnaik, L. M., Mani, V. & Agrawal, V.K. (2000). Application of genetic programming for multicategory pattern classification. *IEEE Transactions on Evolutionary Computation 4*(3), 242-258.

Koza, J.R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, Cambridge, MA: MIT Press.

Koza, J.R. (1994), *Genetic Programming II: Automatic Discovery of Reusable Programs*, Cambridge, MA: MIT Press.

Koza, J.R., Bennett, F. H. III, André, D., & Keane, M. A. (1999), *Genetic Programming III: Darwinian Invention and Problem Solving*. San Mateo, CA: Morgan Kaufmann.

Koza, J.R., Keane, M.A., Streeter, M.J., Mydlowec, W., Yu, J., & Lanza, G. (2003). *Genetic Programming IV: Routine Human-Competitive Machine Intelligence*. San Mateo, CA: Morgan Kaufmann.

Krawiec, K. (2002). Genetic programming-based construction of features for machine learning and knowledge discovery tasks. *Genetic Programming and Evolvable Machines 3*(4), p. 329-343.

Langdon, W. B. & Buxton, B. (2004). Genetic programming for mining DNA chip data from cancer patients. *Genetic Programming and Evolvable Machines*, *5*(3): 251-257, 2004.

Langdon, W.B. & Barrett, S.J. (2005). Genetic programming in data mining for drug discovery. In Ghosh, A. & Jain, L.C., (Eds): *Evolutionary Computing in Data Mining*, pp. 211-235. Studies in Fuzziness and Soft Computing 163. Berlin, Germany: Springer.

Luke, S., Panait, L., Balan, G., Paus, S., Skolicki, Z., Popovici, E., Harrison, J., Bassett, J., Hubley, R., & Chircop, A. (2007). *Evolutionary Computation in Java v*16. Available from URL: http://cs.gmu.edu/~eclab/projects/ecj/.

Luke, S. (2000). *Issues in Scaling Genetic Programming: Breeding Strategies, Tree Generation, and Code Bloat*. Ph.D. Dissertation, Department of Computer Science, University of Maryland, College Park, MD.

Muni, D. P., Pal, N. R. & Das, J. (2004). A novel approach to design classifiers using genetic programming. *IEEE Transactions on Evolutionary Computation 8*(2), 183-196.

Nikolaev, N. Y. & Iba, H. (2001). Regularization approach to inductive genetic programming. *IEEE Transactions on Evolutionary Computation 5*(4), p. 359-375.

Nikolaev, N. Y. & Iba, H. (2001). Accelerated Genetic Programming of Polynomials. *Genetic Programming and Evolvable Machines 2*(3), p. 231-257.

O'Neill, M. & Ryan, C. (2001). Grammatical evolution. *IEEE Transactions on Evolutionary Computation*.

Raymer, M. L., Punch, W. F., & Goodman, E. D., & Kuhn L. A. (1996). Genetic programming for improved data mining - application to the biochemistry of protein interactions. In *Proceedings of the 1st Annual Conference on Genetic Programming*, 375-380. Palo ALto, USA, July 28-31, 1996.

Tsang, C.-H. & Kwong, S. (2007). Ant colony clustering and feature extraction for anomaly intrusion detection. In Grosan, C., Abraham, A., & Chis, M., (Eds): Swarm Intelligence in Data Mining, pp. 101-123. Berlin, Germany: Springer.Brameier, M. & Banzhaf, W. (2001). Evolving Teams of Predictors with Linear Genetic Programming. *Genetic Programming and Evolvable Machines 2*(4), 381-407.

Wikipedia (2007). *Genetic Programming*. Available from URL: http://en.wikipedia.org/wiki/Genetic_programming.

Wong, M. L. & Leung, K. S. (2000). *Data Mining Using Grammar Based Genetic Programming and Applications (Genetic Programming Series, Volume 3)*. Norwell, MA: Kluwer.

KEY TERMS

Automatically-Defined Function (ADF): Parametric functions that are learned and assigned names for reuse as subroutines. ADFs are related to the concept of macro-operators or macros in speedup learning.

Code Growth (Code Bloat): The proliferation of solution elements (e.g., nodes in a tree-based GP representation) that do not contribute towards the objective function.

Crossover: In biology, a process of sexual recombination, by which two chromosomes are paired up

and exchange some portion of their genetic sequence. Crossover in GP is highly stylized and involves structural exchange, typically using subexpressions (subtrees) or production rules in a grammar.

Evolutionary Computation: A solution approach based on simulation models of natural selection, which begins with a set of potential solutions, then iteratively applies algorithms to generate new candidates and select the fittest from this set. The process leads toward a model that has a high proportion of fit individuals.

Generation: The basic unit of progress in genetic and evolutionary computation, a step in which selection is applied over a population. Usually, crossover and mutation are applied once per generation, in strict order.

Individual: A single candidate solution in genetic and evolutionary computation, typically represented using strings (often of fixed length) and permutations in genetic algorithms, or using "problem solver" representations – programs, generative grammars, or circuits – in genetic programming.

Island mode GP: A type of parallel GP where multiple subpopulations (demes) are maintained and evolve independently except during scheduled exchanges of individuals.

Mutation: In biology, a permanent, heritable change to the genetic material of an organism. Mutation in GP involves structural modifications to the elements of a candidate solution. These include changes, insertion, duplication, or deletion of elements (subexpressions, parameters passed to a function, components of a resistor-capacitor-inducer circuit, nonterminals on the right-hand side of a production rule).

Parsimony: An approach in genetic and evolutionary computation, related to "minimum description length", which rewards compact representations by imposing a penalty for individuals in direct proportion to their size (e.g., number of nodes in a GP tree). The rationale for parsimony is that it promotes generalization in supervised inductive learning and produces solutions with less code, which can be more efficient to apply.

Selection: In biology, a mechanism in by which the fittest individuals survive to reproduce, and the basis of speciation according to the Darwinian theory of evolution. Selection in GP involves evaluation of a quantitative criterion over a finite set of fitness cases, with the combined evaluation measures being compared in order to choose individuals.

Genetic Programming for Automatically Constructing Data Mining Algorithms

Alex A. Freitas
University of Kent, UK

Gisele L. Pappa
Federal University of Minas Geras, Brazil

INTRODUCTION

At present there is a wide range of data mining algorithms available to researchers and practitioners (Witten & Frank, 2005; Tan et al., 2006). Despite the great diversity of these algorithms, virtually all of them share one feature: they have been *manually* designed. As a result, current data mining algorithms in general incorporate human biases and preconceptions in their designs.

This article proposes an alternative approach to the design of data mining algorithms, namely the automatic creation of data mining algorithms by means of Genetic Programming (GP) (Pappa & Freitas, 2006). In essence, GP is a type of Evolutionary Algorithm – i.e., a search algorithm inspired by the Darwinian process of natural selection – that evolves computer programs or executable structures.

This approach opens new avenues for research, providing the means to design novel data mining algorithms that are less limited by human biases and preconceptions, and so offer the potential to discover new kinds of patterns (or knowledge) to the user. It also offers an interesting opportunity for the automatic creation of data mining algorithms tailored to the data being mined.

BACKGROUND

An Evolutionary Algorithm (EA) is a computational problem-solving method inspired by the process of natural selection. In essence, an EA maintains a population of "individuals", where each individual represents a candidate solution to the target problem. EAs are iterative generate-and-test procedures, where at each "generation" (iteration) a population of individuals is generated and each individual has its "fitness" computed. The fitness of an individual is a measure of the quality of its corresponding candidate solution. The higher the fitness of an individual, the higher the probability that the individual will be selected to be a "parent" individual. Certain operations (often "crossover" and/or "mutation" operations inspired by their natural counterparts) are applied to the selected parent individuals in order to produce "children" individuals. The important point is that, since the children are in general produced from parents selected based on fitness, the children (new candidate solutions) tend to inherit parts of the good solutions of the previous generation, and the population as a whole gradually evolves to fitter and fitter individuals (better and better solutions to the target problem). For a comprehensive review of EAs in general the reader is referred to (De Jong, 2006; Eiben & Smith, 2003), and a comprehensive review of EAs for data mining can be found in (Freitas, 2002).

The basic principle of EAs – i.e., artificial evolution of candidate solutions, where parent solutions are selected based on their fitness in order to create new children solutions – still holds in Genetic Programming (GP). The main distinguishing feature of GP – by comparison with other types of EA – is that the candidate solutions represented by GP individuals are (or at least should be) computer programs or executable structures. GP has been an active research field for about 15 years (Koza, 1992; Banzhaf et al., 1998; Langdon & Poli, 2002; Koza, 2003), and Koza (2006) reports 36 instances where GP discovered a solution infringing or duplicating some patent, which led Koza to claim that GP is an automated invention machine that routinely produces human-competitive results. However, the creation of a GP system for automatically evolving a full data mining algorithm, as proposed in this article, is a new research topic which is just now

starting to be systematically explored, as discussed in the next section.

MAIN FOCUS

The problem of automatically evolving a data mining algorithm to solve a given data mining task is very challenging, because the evolved algorithm should be generic enough to be applicable to virtually any data set that can be used as input to that task. For instance, an evolved algorithm for the classification task should be able to mine any classification data set (i.e., a data set having a set of records, each of them containing a class attribute and a set of predictor attributes – which can have different data types); an evolved clustering algorithm should be able to mine any clustering data set, etc.

In addition, the automatic evolution of a fully-fledged data mining algorithm requires a GP method that not only is aware of the basic structure of the type of data mining algorithm to be evolved, but also is capable of avoiding fatal programming errors – e.g. avoiding infinite loops when coping with "while" or "repeat" statements. Note that most GP systems in the literature do not have difficulties with the latter problem because they cope only with relatively simple operations (typically mathematical and logical operations), rather than more sophisticated programming statement such as "while" or "repeat" loops. To cope with the two aforementioned problems, a promising approach is to use a grammar-based GP system, as discussed next.

Grammar-Based Genetic Programming

Grammar-Based Genetic Programming (GGP) is a particular type of GP where a grammar is used to create individuals (candidate solutions). There are several types of GGP (Wong & Leung, 2000; O'Neill & Ryan, 2003). A type of GGP particularly relevant for this article consists of individuals that directly encode a candidate program in the form of a tree, namely a derivation tree produced by applying a set of derivation steps of the grammar. A derivation step is simply the application of a production rule to some non-terminal symbol in the left-handed side of the rule, producing the (non-terminal or terminal) symbol in the right-handed side of the rule. Hence, each individual is represented by a derivation tree where the leaf nodes are terminal symbols of the grammar and the internal nodes are the non-terminal symbols of the grammar.

The use of such a grammar is important because it not only helps to constrain the search space to valid algorithms but also guides the GP to exploit valuable background knowledge about the basic structure of the type of data mining algorithm being evolved (Pappa & Freitas, 2006; Wong & Leung, 2000).

In the context of the problem of evolving data mining algorithms the grammar incorporates background knowledge about the type of data mining algorithm being evolved by the GP. Hence, broadly speaking, the non-terminal symbols of the grammar represent high-level descriptions of the major steps in the pseudo-code of a data mining algorithm, whilst the terminal symbols represent a lower-level implementation of those steps.

A New Grammar-Based GP System for Automatically Evolving Rule Induction Algorithms

The previously discussed ideas about Grammar-Based Genetic Programming (GGP) were used to create a GGP system that automatically evolves a rule induction algorithm, guided by a grammar representing background knowledge about the basic structure of rule induction algorithms (Pappa & Freitas, 2006), (Pappa 2007). More precisely, the grammar contains two types of elements, namely:

a. general programming instructions – e.g. *if-then* statements, *for/while* loops; and
b. procedures specifying major operations of rule induction algorithms – e.g., procedures for initializing a classification rule, refining a rule by adding or removing conditions to/from it, selecting a (set of) rule(s) from a number of candidate rules, pruning a rule, etc.

Hence, in this GGP each individual represents a candidate rule induction algorithm, obtained by applying a set of derivation steps from the rule induction grammar. The terminals of the grammar correspond to modular blocks of Java programming code, so each individual is actually a Java program implementing a full rule induction algorithm.

This work can be considered a major "case study" or "proof of concept" for the ambitious idea of automati-

cally evolving a data mining algorithm with genetic programming, and it is further described below. In any case, it should be noted that the proposed idea is generic enough to be applicable to other types of data mining algorithms – i.e., not only rule induction algorithms – with proper modifications in the GGP. The authors' motivation for focusing on rule induction algorithms was that this type of algorithm usually has the advantage of discovering comprehensible knowledge, represented by IF-THEN classification rules (i.e., rules of the form IF (conditions) THEN (predicted class)) that are intuitively interpretable by the user (Witten & Frank, 2005). This is in contrast to some "black box" approaches such as support vector machines or neural networks. Note that the comprehensibility of discovered knowledge is usually recognized as an important criterion in evaluating that knowledge, in the context of data mining (Fayyad et al., 1996; Freitas, 2006).

In order to train the GGP for evolving rule induction algorithms, the authors used a "meta-training set" consisting of six different data sets. The term meta-training set here refers to a set of data sets, all of which are using simultaneously for the training of the GGP – since the goal is to evolve a generic rule induction algorithm. This is in contrast with the normal use of a GP system for rule induction, where the GP is trained with a single data set – since the goal is to evolve just a rule set (rather than algorithm) specific to the given data set. The fitness of each individual – i.e., each candidate rule induction algorithm – is, broadly speaking, an aggregated measure of the classification accuracy of the rule induction algorithm over all the six data sets in the meta-training set. After the evolution is over, the best rule induction algorithm generated by the GGP was then evaluated on a "meta-test set" consisting of five data sets, none of which were used during the training of the GGP. The term meta-test set here refers to a set of data sets, all of which are used to evaluate the generalization ability of a rule induction algorithm evolved by the GGP. Overall, cross-validation results showed that the rule induction algorithms evolved by the GGP system were competitive with several well-known rule induction algorithms – more precisely, CN2, Ripper and C4.5Rules (Witten & Frank, 2005) – which were manually designed and refined over decades of research.

Related Work on GP Systems for Evolving Rule Induction Algorithms

The authors are aware of only two directly relevant related works, as follows. Suyama et al. (1998) proposed a GP system to evolve a classification algorithm. However, the GP system used an ontology consisting of coarse-grained building blocks, where a leaf node of the ontology is a full classification algorithm, and most nodes in the ontology refer to changes in the dataset being mined. Hence, this work can be considered as evolving a good combination of a modified dataset and a classification algorithm selected (out of predefined algorithms) for that modified dataset. By contrast, the methods proposed in (Pappa & Freitas, 2006) are based on a grammar combining finer-grained components of classification algorithms, i.e., the building blocks include programming constructs such as "while" and "if" statements, search strategies, evaluation procedures, etc., which were not used in (Suyama et al., 1998).

In addition, Wong (1998) proposed a grammar-based GP to evolve a classification algorithm adapted to the data being mined. However, that work focused on just one existing rule induction algorithm (viz. FOIL) of which the GP evolved only one component, namely its evaluation function. By contrast, the method proposed in (Pappa & Freitas, 2006) had access to virtually all the components of a classification algorithm, which led to the construction of new rule induction algorithms quite different from existing ones in some runs of the GP.

FUTURE TRENDS

The discussion on the previous section focused on the automatic evolution of rule induction algorithms for classification, but of course this is by no means the only possibility. In principle other types of data mining algorithms can also be automatically evolved by using the previously described approach, as long as a suitable grammar is created to guide the search of the Genetic Programming system. Since this kind of research can be considered pioneering, there are innumerous possibilities for further research varying the type of data mining algorithm to be evolved – i.e., one could evolve other types of classification algorithms, clustering algorithms, etc.

CONCLUSION

This article proposed the idea of designing a Grammar-based Genetic Programming (GGP) system to automatically evolve a data mining algorithm, and briefly discussed, as a practical example of the effectiveness of this approach, a new GGP system that automatically creates rule induction algorithms. In this GGP system the evaluation of candidate rule induction algorithms was performed by using a "meta-training set" consisting of six datasets, in order to generate rule induction algorithms robust across a number of different datasets. This robustness was indeed obtained, since the automatically-evolved rule induction algorithms were competitive with several well-known manually-designed rule induction algorithms in five datasets used in the "meta-test set", representing datasets unseen during the evolution of the GGP.

An alternative research direction, whose experiments are ongoing, is to use a meta-training and a meta-test set having just one dataset. In this approach a subset of the single data set being mined is used in the meta-training set during the GGP evolution, and the remaining subset of the data (i.e., the set of records or data instances not used in the meta-training set) is used in the meta-test set, to evaluate the generalization ability of the rule induction algorithm output by the GGP. In this case the GGP works in a way that actually evolves a rule induction algorithm tailored to the data being mined, which offers the potential to automatically construct data mining algorithms customized to any dataset provided by the user. Note that this represents a new level of automation in data mining. The potentially significant benefit of this automation is clear because the number of datasets to be mined (as well as the number of users) is much greater than the number of (human) data mining algorithm designers, and so one can hardly expect that human designers would have time to manually design algorithms customized to the data being mined.

REFERENCES

Banzhaf, W., Nordin, P., Keller, R.E. and Francone, F.D. (1998) *Genetic Programming – an Introduction: on the automatic evolution of computer programs and its applications*. Palo Alto, CA: Morgan Kaufmann.

De Jong, K. (2006). *Evolutionary Computation: a unified approach.* MIT Press.

Eiben, E.A. & Smith, J.E. (2003). *Introduction to Evolutionary Computation*. Berlin: Springer.

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From data mining to knowledge discovery: an overview. In U.M Fayyad, G. Piatetsky-Shapiro, P. Smyth and R. Uthurusamy (Eds.), *Advances in Knowledge Discovery and Data Mining* (pp. 1-34). Palo Alto, CA: AAAI/MIT.

Freitas, A.A. (2002) *Data Mining and Knowledge Discovery with Evolutionary Algorithms*. Berlin: Springer.

Freitas, A.A. (2006) Are we really discovering "interesting" knowledge from data? *Expert Update, 9(1), Autumn 2006*, 41-47.

Koza, J.R. (1992). *Genetic Programming: on the programming of computers by means of natural selection*. Boston, MA: MIT Press.

Koza, J.R. (2003). *Genetic Programming IV: routine human-competitive machine intelligence*. Berlin: Springer.

Koza, J.R. (2007) http://www.genetic-programming.org/, visited in May 2007.

Langdon, W.B. & Poli, R. (2002). *Foundations of Genetic Programming*. Berlin: Springer.

O'Neill, M. & Ryan, C. (2003). *Grammatical Evolution: Evolutionary Automatic Programming in Arbitrary Language*. Amsterdam: Kluwer.

Pappa, G.L. & Freitas, A.A. (2006). Automatically evolving rule induction algorithms. In *Proc. 17th European Conf. on Machine Learning (ECML-2006), Lecture Notes in Artificial Intelligence, No. 4212* (pp. 341-352). Berlin: Springer.

Pappa, G.L. (2007) Automatically evolving rule induction algorithms with grammar-based genetic programming. *PhD Thesis*. Computing Laboratory, University of Kent, UK.

Suyama, A., Negishi, N. and Yamaguchi, T. (1998). CAMLET: a platform for automatic composition of inductive learning systems using ontologies. In *Proc. Pacific Rim Int. Conf. on AI*, 205-215.

Tan, P.-N., Steinbach, M. and Kumar, V. (2006). *Introduction to Data Mining*. Addison-Wesley.

Witten, I.H. & Frank, E. (2005). *Data Mining: practical machine learning tools and techniques*, 2nd Ed. San Francisco, CA: Morgan Kaufmann.

Wong, M.L. (1998). An adaptive knowledge-acquisition system using genetic programming. *Expert Systems with Applications 15 (1998)*, 47-58.

Wong, M.L. & Leung, K.S. (2000). *Data Mining Using Grammar-Based Genetic Programming and Applications*. Amsterdam: Kluwer.

KEY TERMS

Classification: A type of data mining task where the goal is essentially to predict the class of a record (data instance) based on the values of the predictor attributes for that example. The algorithm builds a classifier from the training set, and its performance is evaluated on a test set (see the definition of training set and test set below).

Classification Rule: A rule of the form IF (conditions) THEN (predicted class), with the meaning that, if a record (data instance) satisfies the conditions in the antecedent of the rule, the rule assigns to that record the class in the consequent of the rule.

Evolutionary Algorithm: An iterative computational problem-solving method that evolves good solutions to a well-defined problem, inspired by the process of natural selection.

Genetic Programming: A type of evolutionary algorithm where the solutions being evolved by the algorithm represent computer programs or executable structures.

Rule Induction Algorithm: A type of data mining algorithm that discovers rules in data.

Test Set: A set of records whose class is unknown by a classification algorithm, used to measure the predictive accuracy of the algorithm after it is trained with records in the training set.

Training Set: A set of records whose class is known, used to train a classification algorithm.

Global Induction of Decision Trees

Marek Kretowski
Bialystok Technical University, Poland

Marek Grzes
Bialystok Technical University, Poland

INTRODUCTION

Decision trees are, besides decision rules, one of the most popular forms of knowledge representation in Knowledge Discovery in Databases process (Fayyad, Piatetsky-Shapiro, Smyth & Uthurusamy, 1996) and implementations of the classical decision tree induction algorithms are included in the majority of data mining systems. A hierarchical structure of a tree-based classifier, where appropriate tests from consecutive nodes are subsequently applied, closely resembles a human way of decision making. This makes decision trees natural and easy to understand even for an inexperienced analyst. The popularity of the decision tree approach can also be explained by their ease of application, fast classification and what may be the most important, their effectiveness.

Two main types of decision trees can be distinguished by the type of tests in non-terminal nodes: *univariate* and *multivariate* decision trees. In the first group, a single attribute is used in each test. For a continuous-valued feature usually an inequality test with binary outcomes is applied and for a nominal attribute mutually exclusive groups of attribute values are associated with outcomes. As a good representative of univariate inducers, the well-known C4.5 system developed by Quinlan (1993) should be mentioned.

In univariate trees a split is equivalent to partitioning the feature space with an axis-parallel hyper-plane. If decision boundaries of a particular dataset are not axis-parallel, using such tests may lead to an over-complicated classifier. This situation is known as the "staircase effect". The problem can be mitigated by applying more sophisticated multivariate tests, where more than one feature can be taken into account. The most common form of such tests is an oblique split, which is based on a linear combination of features (hyper-plane). The decision tree which applies only oblique tests is often called *oblique* or *linear*, whereas heterogeneous trees with univariate, linear and other multivariate (e.g., instance-based) tests can be called *mixed decision trees* (Llora & Wilson, 2004). It should be emphasized that computational complexity of the multivariate induction is generally significantly higher than the univariate induction. CART (Breiman, Friedman, Olshen & Stone, 1984) and OC1 (Murthy, Kasif & Salzberg, 1994) are well known examples of multivariate systems.

BACKGROUND

The issue of finding an optimal decision tree for a given classification problem is known to be a difficult optimization task. Naumov (1991) proved that optimal decision tree construction from data is NP-complete under a variety of measures. In this situation it is obvious that a computationally tractable induction algorithm has to be heuristically enhanced. The most popular strategy is based on the top-down approach (Rokach & Maimon, 2005), where a locally optimal search for tests (based, e.g., on a Gini, towing or entropy rule) and data splitting are recursively applied to consecutive subsets of the training data until the stopping condition is met. Usually, the growing phase is followed by post-pruning (Esposito, Malerba & Semeraro, 1997) aimed at increasing generalization power of the obtained classifier and mitigating the risk of the over-fitting to the learning data.

There are problems where the greedy search fails (e.g., the classical chess board problem) and more sophisticated methods are necessary. In this chapter, we present a global approach, where the whole tree (i.e., its structure and all splits) is constructed at the same time. The motivation for this is the fact that top-down induction with, e.g., entropy minimization, makes locally optimal decisions and at least more compact tree can be obtained when it is constructed and assessed in a global way.

As a first step toward global induction, limited look-ahead algorithms were proposed (e.g., Alopex Perceptron Decision Tree of Shah & Sastry (1999) evaluates quality of a split based on the degree of linear separability of sub-nodes). Another approach consists in a two-stage induction, where a greedy algorithm is applied in the first stage and then the tree is refined to be as close to optimal as possible (GTO (Bennett, 1994) is an example of a linear programming based method for optimizing trees with fixed structures).

In the field of evolutionary computations, the global approach to decision tree induction was initially investigated in genetic programming (GP) community. The tree-based representation of solutions in a population makes this approach especially well-suited and easy for adaptation to decision tree generation. The first attempt was made by Koza (1991), where he presented GP-method for evolving LISP S-expressions corresponding to decision trees. Next, univariate trees were evolved by Nikolaev and Slavov (1998) and Tanigawa and Zhao (2000), whereas Bot and Langdon (2000) proposed a method for induction of classification trees with limited oblique splits.

Among genetic approaches for univariate decision tree induction two systems are particularly interesting here: GATree proposed by Papagelis and Kalles (2001) and GAIT developed by Fu, Golden, Lele, Raghavan and Wasil (2003). Another related global system is named GALE (Llora & Garrell, 2001). It is a fine-grained parallel evolutionary algorithm for evolving both axis-parallel and oblique decision trees.

MAIN FOCUS

We now discuss how evolutionary computation can be applied to induction of decision trees. General concept of a standard evolutionary algorithm is first presented and then we will discuss how it can be applied to build a decision tree classifier.

Evolutionary Algorithms

Evolutionary algorithms (Michalewicz, 1996) belong to a family of metaheuristic methods which represent techniques for solving a general class of difficult computational problems. They provide a general framework (see Figure 1) which is inspired by biological mechanisms of evolution. A biological terminology is used here. The algorithm operates on individuals which compose a current population. Individuals are assessed using a measure named the fitness function and those with higher fitness have usually bigger probability of being selected for reproduction. Genetic operators such as mutation and crossover influence new generations of individuals. This guided random search (offspring usually inherits some traits from its ancestors) is stopped when some convergence criteria is satisfied.

Figure 1. A general framework of evlutionary algorithms

```
(1) Initialize the population
(2) Evaluate initial population
(3) Repeat
    (3.1) Perform competitive selection
    (3.2) Apply genetic operators to generate new solutions
    (3.3) Evaluate solutions in the population
  Until some convergence criteria is satisfied
```

A user defined adaptation of such a general evolutionary algorithm can in itself have heuristic bias (Aguilar-Ruiz, Giráldez, & Riquelme, 2007). It can prune the search space of the particular evolutionary application. The next section shows how this framework was adapted to the problem of decision tree induction.

Decision Tree Induction with Evolutionary Algorithms

In this section the synergy of evolutionary algorithms which are designed to solve difficult computational problems and decision trees which have an NP-complete solution space is introduced. To apply a general algorithm presented in Figure 1 to decision tree induction, the following factors need to be considered:

- Representation of individuals,
- Genetic operators: mutation and crossover,
- Fitness function.

Representation

An evolutionary algorithm operates on individuals. In the evolutionary approach to decision tree induction which is presented in this chapter, individuals are represented as actual trees, which can have different structure and different content. Each individual encoded

as a decision tree represents a potential solution to the problem. Genetic operators operate on such structures and additional post-evolutionary operators ensure that obtained descendant represents a "rational" solution. The problem is that the results of a crossover operator can contain a significant part (sub-tree) which does not have any observations. By pruning or improving those useless parts of obtained individuals the search space can be reduced significantly. In this way such improving actions together with direct representation of individuals in a natural tree like structure bring a user defined heuristic to the evolutionary algorithm.

There are three possible test types which can be used in internal nodes of decision trees discussed in this chapter: two univariate and one multivariate (Kretowski & Grzes, 2007b). In case of univariate tests, a test representation depends on the considered attribute type. For nominal attributes at least one attribute value is associated with each branch. It means that an inner disjunction is implemented. For continuous-valued features typical inequality tests with two outcomes are used. In order to speed up the search process only boundary thresholds (a boundary threshold for the given attribute is defined as a midpoint between such a successive pair of examples in the sequence sorted by the increasing value of the attribute, in which the examples belong to two different classes) are considered as potential splits and they are calculated before starting the evolutionary computation. Finally, an oblique test with a binary outcome can also be applied as a multivariate test (Kretowski & Grzes, 2006).

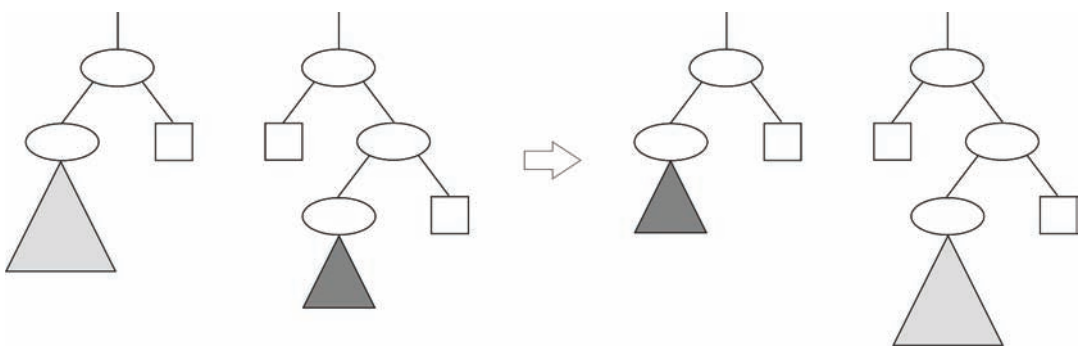
Genetic Operators

Genetic operators are the two main forces that form the basis of evolutionary systems and provide a necessary diversity and novelty. There are two specialized genetic operators corresponding to the classical mutation and crossover. The application of these operators can result in changes of both the tree structure and tests in non-terminal nodes.

Crossover in evolutionary algorithms has its analogy in nature. It is inspired by sexual reproduction of living organisms. The crossover operator attempts to combine elements of two existing individuals (parents) in order to create a new (novel) solution but with some features of its "parents". Because the algorithm is supposed to look for both the structure of the tree and type and parameters of tests, several variants of crossover operators can be applied (Kretowski & Grzes, 2007b). All of them start with selecting the crossover positions – nodes – in two affected individuals. In the most straightforward variant, the subtrees starting in the selected nodes are exchanged. This corresponds to the classical crossover from genetic programming (Koza, 1991). In the second variant, which can be applied only when non-internal nodes have been randomly chosen and the numbers of their outcomes are equal, tests associated with these nodes are exchanged (see Figure 2). The last variant is applicable when non-internal nodes have been drawn and numbers of their descendants are equal. In this case branches which start from the selected nodes are exchanged in random order.

Mutation is also inspired by nature that is by mutation of an organism's DNA. This operator makes periodically random changes in some places of some members of the population. In presented approach, the problem specific representation requires specialized operators. Hence modifications performed by the mutation operator depend for example on the node type (i.e., if the considered node is a leaf node or an internal node). For a non-terminal node a few possibilities exist:

Figure 2. The crossover operator: exchanging sub-trees



- A completely new test of the same or different type can be selected,
- The existing test can be modified,
- An oblique test can be simplified to the corresponding inequality test by eliminating (zeroing) the smallest coefficients or an axis-parallel test can be transformed into oblique one,
- One sub-tree can be replaced by another sub-tree from the same node,
- A node can be transformed (pruned) into a leaf.

Modifying a leaf makes sense only if it contains objects from different classes. The leaf is transformed into an internal node and a new test is randomly chosen. The search for effective tests can be recursively repeated for all newly created descendants (leaves). As a result the mutated leaf can be replaced by a sub-tree, which potentially accelerates the evolution.

Fitness Function

The two forces of evolutionary systems which bring diversity and novelty are crossover and mutation. Selection acts as a force which increases the quality of the population. For this reason the mechanism of selection usually requires a quantitative measure to assess individuals in the population. Such an objective function is called a fitness function and in practice is a very important and sensitive element of the evolutionary system. It drives the learning process in a particular direction. In case of decision trees we need for example to balance the reclassification quality and the complexity (at least size) of the tree. This comes directly from a 14th century principle called Occam's razor which states that an explanation for any phenomenon should take as few assumptions as possible. This has straightforward implications in classification methods, and is particularly noticeable in decision trees. The decision tree which is too big becomes over-fitted to learning data leading to a poor performance on unseen, testing observations.

In a typical top-down induction of decision trees over-fitting is mitigated by defining a stopping condition or by applying a post-pruning. In the discussed approach, the search for an optimal structure is embedded into the evolutionary algorithm by incorporating a complexity term in the fitness function. A similar idea is used in cost complexity pruning in the CART system (Breiman *et al.*, 1984). The fitness function is maximized and has the following form:

$$Fitness(T) = Q_{\mathrm{Re}class}(T) - \alpha(Comp(T) - 1.0)$$

where $Q_{Reclass}(T)$ is the training accuracy (reclassification quality) of the tree T and α is the relative importance of the classifier complexity. In the simplest form the tree complexity $Comp(T)$ can be defined as the classifier size which is usually equal to the number of nodes. The penalty associated with the classifier complexity increases proportionally with the tree size and prevents over-fitting. Subtracting 1.0 eliminates the penalty when the tree is composed of only one leaf (in majority voting). This general form of the fitness function can be adjusted to different requirements like for example induction of mixed decision trees (Kretowski & Grzes, 2007b) which can contain all tree types of tests or put emphasis on the feature selection (Kretowski & Grzes, 2006).

Algorithm Properties

Because the algorithm presented here is built on top of a general metaheuristic concept, it gains flexibility from it. Metaheuristic is a method for solving a general class of problems and because of that it is relatively easy to adjust the algorithm to different goals and demands. Especially the fitness function allows modifying the algorithm in a simple and straightforward way. As it was mentioned in the previous section, the complexity term can take into account the structure of tests in internal nodes and provide feature selection (Kretowski & Grzes, 2006b) when less complicated tests are rewarded.

Our discussion has focused so far on the most popular criterion of classification systems: the prediction error. In many fields, like for example medicine or finance, this approach may not be sufficient when costs of features (due to costly measurements) or misclassification-cost (due to costs of wrong decision) should be taken into account. The term cost-sensitive classification is used in the field (Turney, 1995). The algorithm was adjusted to meet also these criteria. The modification of the fitness function allowed selecting features in accordance to their cost and optimizing the misclassification cost instead of the prediction error (Kretowski & Grzes, 2007a).

FUTURE TRENDS

While investigating evolutionary algorithms there is always a strong motivation for speeding them up. It is also a well-known fact that evolutionary approaches are well suited for parallel architecture (Alba, 2005) and an implementation based on Message Passing Interface (MPI) is available. This is especially important in the context of modern data mining applications, where huge learning sets need to be analyzed.

Another possibility of speeding up and focusing the evolutionary search is embedding the local search operators into the algorithm (so called *memetic algorithms*). Applications of such hybrid solutions to the global induction are currently investigated (Kretowski, 2008).

CONCLUSION

In this chapter, the global induction of univariate, oblique and mixed decision trees is presented. In contrast to the classical top-down approaches, both the structure of the classifier and all tests in internal nodes are searched during one run of the evolutionary algorithm. Specialized genetic operators applied in informed way and suitably defined fitness function allow us to focus the search process and efficiently generate competitive classifiers also for cost-sensitive problems. Globally induced trees are generally less complex and in certain situations more accurate.

REFERENCES

Aguilar-Ruiz, J. S., Giráldez, R., & Riquelme J. C. (2007). Natural Encoding for Evolutionary Supervised Learning. IEEE Transactions on Evolutionary Computation, 11(4), 466-479.

Alba, E. (2005). Parallel Metaheuristics: A New Class of Algorithms. Wiley-Interscience.

Bennett, K. (1994). Global tree optimization: A non-greedy decision tree algorithm. Computing Science and Statistics, 26, 156-160.

Bot, M., & Langdon, W. (2000). Application of genetic programming to induction of linear classification trees. Proceedings of EuroGP'2000, Springer, LNCS 1802, 247-258.

Breiman, L., Friedman, J., Olshen, R., & Stone C. (1984). Classification and regression trees. Wadsworth International Group.

Brodley, C. (1995). Recursive automatic bias selection for classifier construction. Machine Learning, 20, 63-94.

Cantu-Paz, E., & Kamath, C. (2003). Inducing oblique decision trees with evolutionary algorithms. IEEE Transactions on Evolutionary Computation, 7(1), 54-68.

Esposito, F., Malerba, D., & Semeraro, G. (1997). A comparative analysis of methods for pruning decision trees. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(5), 476-491.

Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., & Uthurusamy, R. (1996). Advances in Knowledge Discovery and Data Mining, AAAI Press.

Fu, Z., Golden, B., Lele, S., Raghavan, S., & Wasil, E. (2003). A genetic algorithm based approach for building accurate decision trees. INFORMS Journal of Computing, 15(1), 3-22.

Koza, J. (1991). Concept formation and decision tree induction using genetic programming paradigm. Proceedings of International Conference on Parallel Problem Solving from Nature - Proceedings of 1st Workshop, Springer, LNCS 496, 124-128.

Kretowski, M., & Grzes, M. (2006). Evolutionary learning of linear trees with embedded feature selection. Proceedings of International Conference on Artificial Intelligence and Soft Computing, Springer, LNCS 4029.

Kretowski, M., & Grzes, M. (2007a). Evolutionary Induction of Decision Trees for Misclassification Cost Minimization. Proceedings of 8th International Conference on Adaptive and Natural Computing Algorithms, Springer, LNCS 4431, 1-10.

Kretowski, M., & Grzes, M. (2007b). Evolutionary Induction of Mixed Decision Trees. International Journal of Data Warehousing and Mining, 3(4), 68-82.

Kretowski, M. (2008). A Memetic Algorithm for Global Induction of Decision Trees. Proceedings of 34th International Conference on Current Trends in Theory and Practice of Computer, Springer LNCS, 4910, 531-540.

Llora, X., Garrell, J. (2001). Evolution of decision trees. Proceedings of CCAI'01, ACIA Press, 115-122.

Llora, X., & Wilson, S. (2004). Mixed decision trees: Minimizing knowledge representation bias in LCS. Proceedings of Genetic and Evolutionary Computation Conference, Springer, LNCS 3103, 797-809.

Michalewicz, Z. (1996). Genetic algorithms + data structures = evolution programs (3rd ed.). Springer.

Murthy, S., Kasif, S., & Salzberg, S. (1994). A system for induction of oblique decision trees. Journal of Artificial Intelligence Research, 2, 1-33.

Naumov, G. E. (1991). NP-completeness of problems of construction of optimal decision trees. Soviet Physics Doklady, 36(4), 270-271.

Nikolaev, N., & Slavov, V. (1998). Inductive genetic programming with decision trees. Intelligent Data Analysis, 2, 31-44.

Papagelis, A., & Kalles, D. (2001). Breeding decision trees using evolutionary techniques. Proceedings of International Conference on Machine Learning, Morgan Kaufmann, 393-400.

Rokach, L., & Maimon, O. (2005). Top-down induction of decision trees classifiers - A survey. IEEE Transactions On Systems, Man, and Cybernetics - Part C, 35(4), 476-487.

Quinlan, J. (1993). C4.5: Programs for machine learning. Morgan Kaufmann.

Shah, S. & Sastry, P. S. (1999). New Algorithms for Learning and Pruning Oblique Decision Trees. IEEE Transactions On Systems, Man, And Cybernetics, 29(4).

Tanigawa, T. & Zhao, Q. (2000). A Study on Efficient Generation of Decision Trees Using Genetic Programming. Proceedings of the Genetic and Evolutionary Computation Conference, 1047-1052.

Turney, P. (1995). Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm. Journal of Artificial Intelligence Research, 2, 369–409.

KEY TERMS

Cost-Sensitive Classifier: A classifier, which can optimize at least one type of cost: misclassification cost (assigning different costs to different types of misclassifications) or cost of features (values of features usually come from measurements of different costs).

Decision Tree Pruning: The approach to avoiding over-fitting the input data. Firstly, the tree which over-fits the data is constructed and then pruning simplifies this tree by pruning its lower parts. This process generally aims at obtaining the simplest tree with the highest classification of input data that is at improving the generalization power of the decision tree.

Global Induction: A method of decision tree generation, where both the tree structure and all tests are searched at the same time; usually based on evolutionary approach in contrast to top-down induction.

Mixed Decision Trees: A decision tree in which different types of tests are permitted: univariate (e.g., nominal and inequality tests) and multivariate (e.g., oblique or instance-based tests).

Oblique Decision Tree: A binary decision tree with tests based on hyper-planes (linear combinations of features).

Over-Fitting: The problem existing in supervised learning when a classifier perfectly classifies the training data but performs much worse on unseen data; the problem can emerge when machine learning algorithm fits noise or insufficient data, i.e., when it learns on irrelevant facts or generalizes from specific cases.

Top-Down Induction: A recursive method of decision tree generation; starts with the entire input dataset in the root node; an optimal test for data splitting is searched and branches corresponding to the test outcomes are created; different measures for test evaluation have been used, e.g., Gini, entropy and towing rules; test searches and data splitting are repeated in the created nodes unless the stopping condition is met.

Graph-Based Data Mining

Lawrence B. Holder
University of Texas at Arlington, USA

Diane J. Cook
University of Texas at Arlington, USA

INTRODUCTION

Graph-based data mining represents a collection of techniques for mining the relational aspects of data represented as a graph. Two major approaches to graph-based data mining are *frequent subgraph mining* and *graph-based relational learning*. This chapter will focus on one particular approach embodied in the Subdue system, along with recent advances in graph-based supervised learning, graph-based hierarchical conceptual clustering, and graph-grammar induction.

Most approaches to data mining look for associations among an entity's attributes, but relationships between entities represent a rich source of information, and ultimately knowledge. The field of *multi-relational data mining*, of which graph-based data mining is a part, is a new area investigating approaches to mining this relational information by finding associations involving multiple tables in a relational database. Two main approaches have been developed for mining relational information: logic-based approaches and graph-based approaches.

Logic-based approaches fall under the area of *inductive logic programming* (ILP). ILP embodies a number of techniques for inducing a logical theory to describe the data, and many techniques have been adapted to multi-relational data mining (Dzeroski & Lavrac, 2001; Dzeroski, 2003). Graph-based approaches differ from logic-based approaches to relational mining in several ways, the most obvious of which is the underlying representation. Furthermore, logic-based approaches rely on the prior identification of the predicate or predicates to be mined, while graph-based approaches are more data-driven, identifying any portion of the graph that has high support. However, logic-based approaches allow the expression of more complicated patterns involving, e.g., recursion, variables, and constraints among variables. These representational limitations of graphs can be overcome, but at a computational cost.

BACKGROUND

Graph-based data mining (GDM) is the task of finding novel, useful, and understandable graph-theoretic patterns in a graph representation of data. Several approaches to GDM exist based on the task of identifying frequently occurring subgraphs in graph transactions, i.e., those subgraphs meeting a minimum level of support. Washio and Motoda (2003) provide an excellent survey of these approaches. We here describe four representative GDM methods.

Kuramochi and Karypis (2001) developed the FSG system for finding all frequent subgraphs in large graph databases. FSG starts by finding all frequent single and double edge subgraphs. Then, in each iteration, it generates candidate subgraphs by expanding the subgraphs found in the previous iteration by one edge. In each iteration the algorithm checks how many times the candidate subgraph occurs within an entire graph. The candidates, whose frequency is below a user-defined level, are pruned. The algorithm returns all subgraphs occurring more frequently than the given level.

Yan and Han (2002) introduced gSpan, which combines depth-first search and lexicographic ordering to find frequent subgraphs. Their algorithm starts from all frequent one-edge graphs. The labels on these edges together with labels on incident vertices define a code for every such graph. Expansion of these one-edge graphs maps them to longer codes. Since every graph

can map to many codes, all but the smallest code are pruned. Code ordering and pruning reduces the cost of matching frequent subgraphs in gSpan. Yan and Han (2003) describe a refinement to gSpan, called CloseGraph, which identifies only subgraphs satisfying the minimum support, such that no supergraph exists with the same level of support.

Inokuchi et al. (2003) developed the Apriori-based Graph Mining (AGM) system, which searches the space of frequent subgraphs in a bottom-up fashion, beginning with a single vertex, and then continually expanding by a single vertex and one or more edges. AGM also employs a canonical coding of graphs in order to support fast subgraph matching. AGM returns association rules satisfying user-specified levels of support and confidence.

The last approach to GDM, and the one discussed in the remainder of this chapter, is embodied in the Subdue system (Cook & Holder, 2000). Unlike the above systems, Subdue seeks a subgraph pattern that not only occurs frequently in the input graph, but also significantly compresses the input graph when each instance of the pattern is replaced by a single vertex. Subdue performs a greedy search through the space of subgraphs, beginning with a single vertex and expanding by one edge. Subdue returns the pattern that maximally compresses the input graph. Holder and Cook (2003) describe current and future directions in this graph-based relational learning variant of GDM.

MAIN THRUST OF THE CHAPTER

As a representative of GDM methods, this section will focus on the Subdue graph-based data mining system. The input data is a directed graph with labels on vertices and edges. Subdue searches for a substructure that best compresses the input graph. A *substructure* consists of a subgraph definition and all its occurrences throughout the graph. The initial state of the search is the set of substructures consisting of all uniquely labeled vertices. The only operator of the search is the *Extend Substructure* operator. As its name suggests, it extends a substructure in all possible ways by a single edge and a vertex, or by only a single edge if both vertices are already in the subgraph.

Subdue's search is guided by the *minimum description length* (MDL) principle, which seeks to minimize the description length of the entire data set. The evaluation heuristic based on the MDL principle assumes that the best substructure is the one that minimizes the description length of the input graph when compressed by the substructure. The description length of the substructure *S* given the input graph *G* is calculated as $DL(G,S) = DL(S)+DL(G|S)$, where $DL(S)$ is the description length of the substructure, and $DL(G|S)$ is the description length of the input graph compressed by the substructure. Subdue seeks a substructure S that minimizes $DL(G,S)$.

The search progresses by applying the *Extend Substructure* operator to each substructure in the current state. The resulting state, however, does not contain all the substructures generated by the *Extend Substructure* operator. The substructures are kept on a queue and are ordered based on their description length (or sometimes referred to as *value*) as calculated using the MDL principle. The queue's length is bounded by a user-defined constant.

The search terminates upon reaching a user-specified limit on the number of substructures extended, or upon exhaustion of the search space. Once the search terminates and Subdue returns the list of best substructures found, the graph can be compressed using the best substructure. The compression procedure replaces all instances of the substructure in the input graph by single vertices, which represent the substructure's instances. Incoming and outgoing edges to and from the replaced instances will point to, or originate from the new vertex that represents the instance. The Subdue algorithm can be invoked again on this compressed graph.

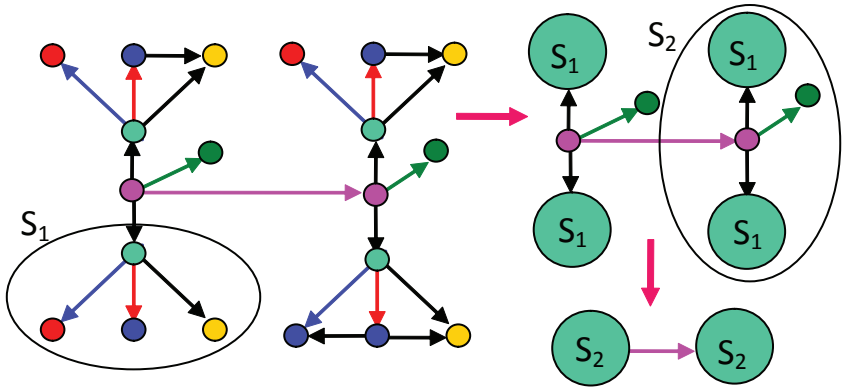
Figure 1 illustrates the GDM process on a simple example. Subdue discovers substructure S_1, which is used to compress the data. Subdue can then run for a second iteration on the compressed graph, discovering substructure S_2. Because instances of a substructure can appear in slightly different forms throughout the data, an inexact graph match, based on graph edit distance, is used to identify substructure instances.

Most GDM methods follow a similar process. Variations involve different heuristics (e.g., frequency vs. MDL) and different search operators (e.g., merge vs. extend).

Graph-Based Hierarchical Conceptual Clustering

Given the ability to find a prevalent subgraph pattern in a larger graph and then compress the graph with this

Figure 1. Graph-based data mining: A simple example.



pattern, iterating over this process until the graph can no longer be compressed will produce a hierarchical, conceptual clustering of the input data. On the i^{th} iteration, the best subgraph S_i is used to compress the input graph, introducing new vertices labeled S_i in the graph input to the next iteration. Therefore, any subsequently-discovered subgraph S_j can be defined in terms of one or more of S_is, where $i < j$. The result is a *lattice*, where each cluster can be defined in terms of more than one parent subgraph. For example, Figure 2 shows such a clustering done on a DNA molecule. Note that the ordering of pattern discovery can affect the parents of a pattern. For instance, the lower-left pattern in Figure 2 could have used the C-C-O pattern, rather than the C-C pattern, but in fact, the lower-left pattern is discovered before the C-C-O pattern. For more information on graph-based clustering, see (Jonyer et al., 2001).

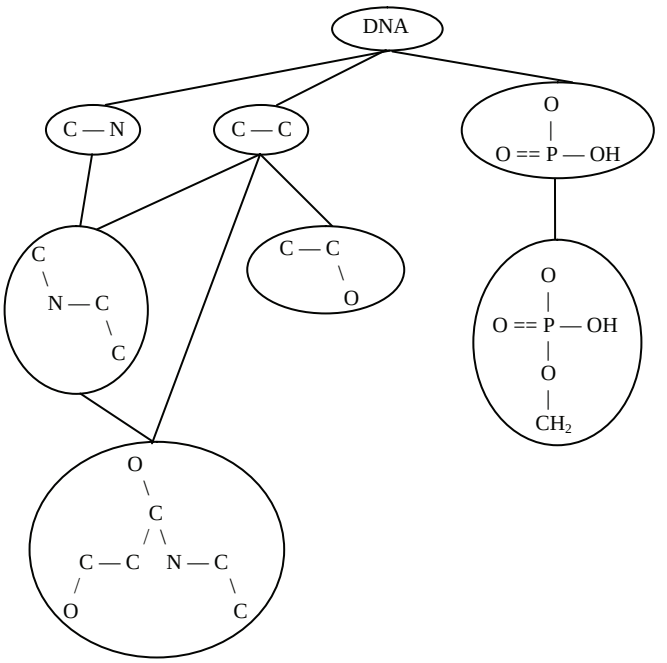
Graph-Based Supervised Learning

Extending a graph-based data mining approach to perform supervised learning involves the need to handle negative examples (focusing on the two-class scenario). In the case of a graph the negative information can come in three forms. First, the data may be in the form of numerous smaller graphs, or graph transactions, each labeled either positive or negative. Second, data may be composed of two large graphs: one positive and one negative. Third, the data may be one large graph in which the positive and negative labeling occurs throughout. We will talk about the third scenario in the section on future directions. The first scenario is closest to the standard supervised learning problem in that we have a set of clearly defined examples. Let G^+ represent the set of positive graphs, and G^- represent the set of negative graphs. Then, one approach to supervised learning is to find a subgraph that appears often in the positive graphs, but not in the negative graphs. This amounts to replacing the information-theoretic measure with simply an *error-based measure*. This approach will lead the search toward a small subgraph that discriminates well. However, such a subgraph does not necessarily compress well, nor represent a characteristic description of the target concept.

We can bias the search toward a more characteristic description by using the information-theoretic measure to look for a subgraph that compresses the positive examples, but not the negative examples. If $I(G)$ represents the description length (in bits) of the graph G, and $I(G|S)$ represents the description length of graph G compressed by subgraph S, then we can look for an S that minimizes $I(G^+|S) + I(S) + I(G^-) - I(G^-|S)$, where the last two terms represent the portion of the negative graph incorrectly compressed by the subgraph. This approach will lead the search toward a larger subgraph that characterizes the positive examples, but not the negative examples.

Finally, this process can be iterated in a set-covering approach to learn a disjunctive hypothesis. If using the error measure, then any positive example containing the learned subgraph would be removed from subsequent iterations. If using the information-theoretic measure, then instances of the learned subgraph in both the positive and negative examples (even multiple instances

Figure 2. Graph-based hierarchical, conceptual clustering of a DNA molecule.



per example) are compressed to a single vertex. Note that the compression is a lossy one, i.e, we do not keep enough information in the compressed graph to know how the instance was connected to the rest of the graph. This approach is consistent with our goal of learning general patterns, rather than mere compression. For more information on graph-based supervised learning, see (Gonzalez et al., 2002).

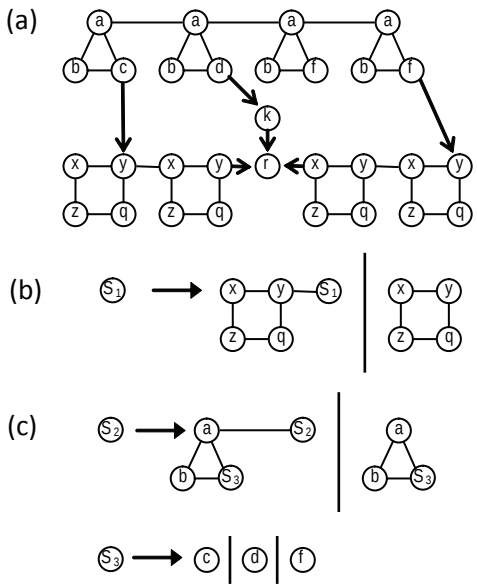
Graph Grammar Induction

As mentioned earlier, two of the advantages of logic-based approach to relational learning are the ability to learn recursive hypotheses and constraints among variables. However, there has been much work in the area of graph grammars, which overcome this limitation. Graph grammars are similar to string grammars except that terminals can be arbitrary graphs rather than symbols from an alphabet. While much of the work on graph grammars involves the analysis of various classes of graph grammars, recent research has begun to develop techniques for learning graph grammars (Doshi et al., 2002; Jonyer et al., 2002).

Figure 3b shows an example of a *recursive graph grammar* production rule learned from the graph in Figure 3a. A GDM approach can be extended to consider graph grammar productions by analyzing the instances of a subgraph to see how they are related to each other. If two or more instances are connected to each other by one or more edges, then a recursive production rule generating an infinite sequence of such connected subgraphs can be constructed. A slight modification to the information-theoretic measure taking into account the extra information needed to describe the recursive component of the production is all that is needed to allow such a hypothesis to compete along side simple subgraphs (i.e., terminal productions) for maximizing compression.

These graph grammar productions can include non-terminals on the right-hand side. These productions can be disjunctive, as in Figure 3c, which represents the final production learned from Figure 3a using this approach. The disjunction rule is learned by looking for similar, but not identical, extensions to the instances of a subgraph. A new rule can be constructed that captures the disjunctive nature of this extension, and included in the pool of production rules competing based on their ability to compress the input graph. With a proper encoding of this disjunction information, the MDL criterion will tradeoff the complexity of the rule with the

Figure 3. Graph grammar learning example with (a) the input graph, (b) the first grammar rule learned, and (c) the second and third grammar rules learned.



amount of compression it affords in the input graph. An alternative to defining these disjunction non-terminals is to instead construct a variable whose range consists of the different disjunctive values of the production. In this way we can introduce constraints among variables contained in a subgraph by adding a constraint edge to the subgraph. For example, if the four instances of the triangle structure in Figure 3a each had another edge to a *c*, *d*, *f* and *f* vertex respectively, then we could propose a new subgraph, where these two vertices are represented by variables, and an equality constraint is introduced between them. If the range of the variable is numeric, then we can also consider inequality constraints between variables and other vertices or variables in the subgraph pattern.

Jonyer (2003) has developed a graph grammar learning approach with the above capabilities. The approach has shown promise both in handling noise and learning recursive hypotheses in many different domains including learning the building blocks of proteins and communication chains in organized crime.

FUTURE TRENDS

The field of graph-based relational learning is still young, but the need for practical algorithms is growing fast. Therefore, we need to address several challenging scalability issues, including incremental learning in dynamic graphs. Another issue regarding practical applications involves the blurring of positive and negative examples in a supervised learning task, that is, the graph has many positive and negative parts, not easily separated, and with varying degrees of class membership.

Partitioning and Incremental Mining for Scalability

Scaling GDM approaches to very large graphs, graphs too big to fit in main memory, is an ever-growing challenge. Two approaches to address this challenge are being investigated. One approach involves *partitioning* the graph into smaller graphs that can be processed in a distributed fashion (Cook et al., 2001). A second

approach involves implementing GDM within a relational database management system, taking advantage of user-defined functions and the optimized storage capabilities of the RDBMS.

A newer issue regarding scalability involves *dynamic graphs*. With the advent of real-time streaming data, many data mining systems must mine incrementally, rather than off-line from scratch. Many of the domains we wish to mine in graph form are dynamic domains. We do not have the time to periodically rebuild graphs of all the data to date and run a GDM system from scratch. We must develop methods to incrementally update the graph and the patterns currently prevalent in the graph. One approach is similar to the graph partitioning approach for distributed processing. New data can be stored in an increasing number of partitions. Information within partitions can be exchanged, or a repartitioning can be performed if the information loss exceeds some threshold. GDM can be used to search the new partitions, suggesting new subgraph patterns as they evaluate highly in new and old partitions.

Supervised Learning with Blurred Graphs

In a highly relational domain the positive and negative examples of a concept are not easily separated. Such a graph is called a *blurred graph*, in that the graph as a whole contains class information, but perhaps not individual components of the graph. This scenario presents a challenge to any data mining system, but especially to a GDM system, where clearly classified data may be tightly related to less classified data. Two approaches to this task are being investigated. The first involves modifying the MDL encoding to take into account the amount of information necessary to describe the class membership of compressed portions of the graph. The second approach involves treating the class membership of a vertex or edge as a cost and weighting the information-theoretic value of the subgraph patterns by the costs of the instances of the pattern. The ability to learn from blurred graphs will allow the user more flexibility in indicating class membership where known, and to varying degrees, without having to clearly separate the graph into disjoint examples.

CONCLUSION

Graph-based data mining (GDM) is a fast-growing field due to the increasing interest in mining the relational aspects of data. We have described several approaches to GDM including logic-based approaches in ILP systems, graph-based frequent subgraph mining approaches in AGM, FSG and gSpan, and a graph-based relational learning approach in Subdue. We described the Subdue approach in detail along with recent advances in supervised learning, clustering, and graph-grammar induction.

However, much work remains to be done. Because many of the graph-theoretic operations inherent in GDM are NP-complete or definitely not in P, scalability is a constant challenge. With the increased need for mining streaming data, the development of new methods for incremental learning from dynamic graphs is important. Also, the blurring of example boundaries in a supervised learning scenario gives rise to graphs, where the class membership of even nearby vertices and edges can vary considerably. We need to develop better methods for learning in these blurred graphs.

Finally, we discussed several domains throughout this paper that benefit from a graphical representation and the use of GDM to extract novel and useful patterns. As more and more domains realize the increased predictive power of patterns in relationships between entities, rather than just attributes of entities, graph-based data mining will become foundational to our ability to better understand the ever-increasing amount of data in our world.

REFERENCES

Cook, D. & Holder, L. (2000). Graph-based data mining. *IEEE Intelligent Systems*, 15(2):32-41.

Cook, D., Holder, L., Galal, G. & Maglothin, R. (2001). Approaches to parallel graph-based knowledge discovery. *Journal of Parallel and Distributed Computing*, 61(3):427-446.

Doshi, S., Huang, F. & Oates, T. (2002). Inferring the structure of graph grammars from data. *Proceedings of the International Conference on Knowledge-based Computer Systems*.

Dzeroski, S. & Lavrac, N. (2001). *Relational Data Mining*. Springer.

Dzeroski, S. (2003). Multi-relational data mining: An introduction. *SIGKDD Explorations*, 5(1):1-16.

Gonzalez, J., Holder, L. & Cook, D. (2002). Graph-based relational concept learning. *Proceedings of the Nineteenth International Conference on Machine Learning*.

Holder, L. & Cook, D. (2003). Graph-based relational learning: Current and future directions. *SIGKDD Explorations*, 5(1):90-93.

Inokuchi, A., Washio, T. & Motoda, H. (2003). Complete mining of frequent patterns from graphs: Mining graph data. *Machine Learning*, 50:321-254.

Jonyer, I., Cook, D. & Holder, L. (2001). Graph-based hierarchical conceptual clustering. *Journal of Machine Learning Research*, 2:19-43.

Jonyer, I., Holder, L. & Cook, D. (2002). Concept formation using graph grammars. *Proceedings of the KDD Workshop on Multi-Relational Data Mining*.

Jonyer, I. (2003). Context-free graph grammar induction using the minimum description length principle. Ph.D. thesis. Department of Computer Science and Engineering, University of Texas at Arlington.

Kuramochi, M. & Karypis, G. (2001). Frequent subgraph discovery. *Proceedings of the First IEEE Conference on Data Mining*.

Washio, T. & Motoda, H. (2003). State of the art of graph-based data mining. *SIGKDD Explorations*, 5(1):59-68.

Yan, X. & Han, J. (2002). Graph-based substructure pattern mining. *Proceedings of the International Conference on Data Mining*.

Yan, X. & Han, J. (2003). CloseGraph: Mining closed frequent graph patterns. *Proceedings of the Ninth International Conference on Knowledge Discovery and Data Mining*.

KEY TERMS

Blurred Graph: Graph in which each vertex and edge can belong to multiple categories to varying degrees. Such a graph complicates the ability to clearly define transactions on which to perform data mining.

Conceptual Graph: Graph representation described by a precise semantics based on first-order logic.

Dynamic Graph: Graph representing a constantly-changing stream of data.

Frequent Subgraph Mining: Finding all subgraphs within a set of graph transactions whose frequency satisfies a user-specified level of minimum support.

Graph-based Data Mining: Finding novel, useful, and understandable graph-theoretic patterns in a graph representation of data.

Graph Grammar: Grammar describing the construction of a set of graphs, where terminals and non-terminals represent vertices, edges or entire subgraphs.

Inductive Logic Programming: Techniques for learning a first-order logic theory to describe a set of relational data.

Minimum Description Length (MDL) Principle: Principle stating that the best theory describing a set of data is the one minimizing the description length of the theory plus the description length of the data described (or compressed) by the theory.

Multi-Relational Data Mining: Mining patterns that involve multiple tables in a relational database.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 540-545, copyright 2006 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

Graphical Data Mining

Carol J. Romanowski
Rochester Institute of Technology, USA

INTRODUCTION

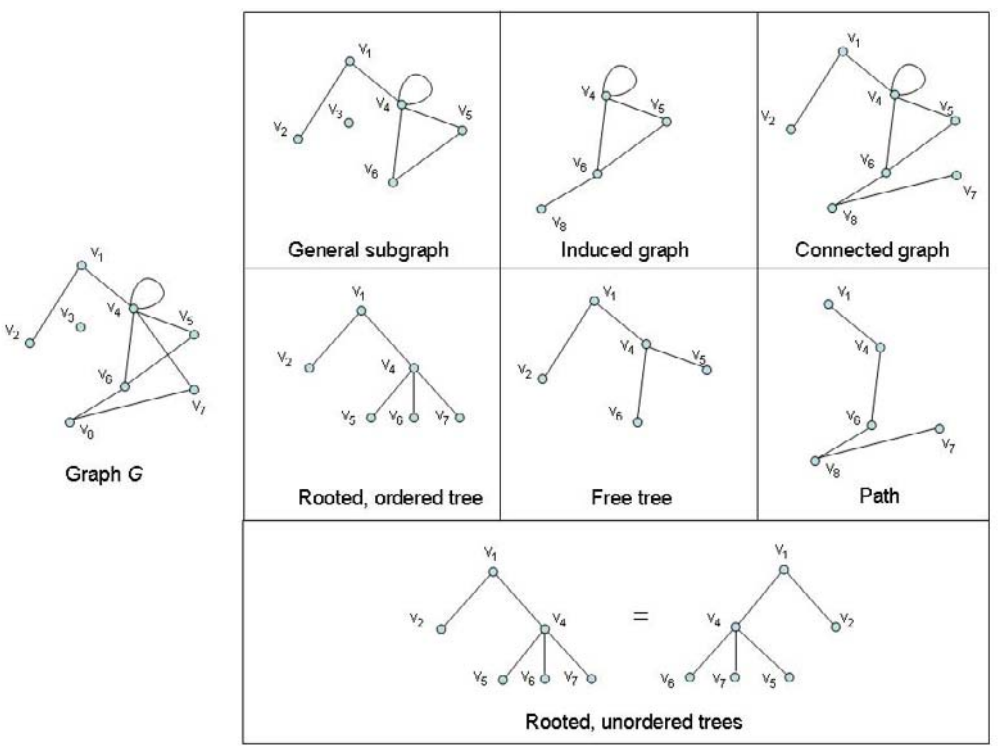
Data mining has grown to include many more data types than the "traditional" flat files with numeric or categorical attributes. Images, text, video, and the internet are now areas of burgeoning data mining research. Graphical data is also an area of interest, since data in many domains—such as engineering design, network intrusion detection, fraud detection, criminology, document analysis, pharmacology, and biochemistry—can be represented in this form.

Graph mining algorithms and methods are fewer and less mature than those designed for numerical or categorical data. In addition, the distinction between graph matching and graph mining is not always clear. In graph mining, we often want to find all possible frequent subgraphs of all possible sizes that occur a specified minimum number of times. That goal involves iteratively matching incrementally larger subgraphs, while classical graph matching is a single search for a static subgraph. Also, graph mining is an unsupervised learning task. Instead of searching for a single match to a specific graph, we are looking for known or unknown graphs embedded in the data.

BACKGROUND

A graph *G* is a structure that contains a set of vertices *V* and their incident edges *E* and comprises many

Figure 1. Graph structures



substructures (see Figure 1). A *general subgraph* is a subset of any vertices and edges in the parent graph. An *induced subgraph* contains a subset of the vertices and all edges between those vertices that exist in the parent graph. A *connected subgraph* is a subset of vertices that are connected with edges; no isolated vertices are allowed. *Trees* are acyclic, directed, branched subgraphs that can be ordered (the order of branch nodes is fixed) or unordered (the order of branch nodes is of no concern). Rooted trees have one vertex with no edges coming into it; all other vertices are reachable from the root, while free trees have no root. A *path* is a subgraph that has no branches.

The most common graph mining task is finding frequent subgraph structures within either a large, single graph or a database of smaller graphs (Washio & Motoda, 2003). Most methods for finding frequent subgraphs are based on the Apriori algorithm (Agrawal & Srikant, 1994) and involve four steps:

1. Starting with a defined smallest unit, generate increasingly larger subgraphs.
2. Check that the generated subgraphs appear in the database or graph.
3. Count the number of times each subgraph appears.
4. Discard subgraphs that are less frequent than the user-specified minimum, or *support*, thus avoiding investigation of their supergraphs. Also discard isomorphisms (subgraphs with identical vertices and edges).

Subgraph matching, or isomorphism testing, is thought to have no polynomial time algorithm for general graphs (Skiena, 1998). Graph mining algorithms attempt to reduce the computational load of this problem in many ways, all of which rely on the downward closure property (see step 4). This property states that a subgraph of a larger graph is more frequent than the larger graph; therefore, if a particular subgraph is infrequent, it is removed from the search space because further expansion would not yield a viable candidate. Another way to reduce search space is to restrict candidates to subgraphs with no common edges (Washio & Motoda, 2003).

MAIN FOCUS

The most basic difference among graph mining algorithms is whether the input data is a single graph or a database of smaller graphs. Algorithms written for a single graph input can be used on a database of smaller graphs, but the reverse is not true (Goethels et al., 2005). Most general graph and tree mining algorithms are frequent subgraph methods focused on a database of graphs, although the single graph approach is more versatile.

Within the main categories of single vs. database of graphs, these frequent subgraph methods are often very similar, and are mostly distinguished by the strategies for generating candidate subgraphs and reducing the support computation cost, by the method of graph matching employed, and by the basic substructure unit. Complete algorithms generate every possible candidate subgraph, while heuristic algorithms truncate the candidate generation stage.

Many of the complete methods such as AGM and AcGM (Inokuchi et al., 2000, 2003), FSG (Kuramochi & Karypis, 2004), GASTON (Nijssen & Kok, 2004) and algorithms proposed by Vanetik et al. (2004) and Inokuchi (2004) use the join method of Apriori to generate new candidate subgraphs. In this method, two graphs of size k with identical subgraphs of size k-1are joined together to form a graph of size k+1. For example, consider two connected graphs, each with k = 3 nodes. Graph A contains nodes {B, E, and K} while Graph B contains nodes {C, E, and K}. The new candidate graph would contain nodes {B, C, E, and K} and their incident edges.

Each of these algorithms uses a different substructure unit (either a vertex, edge, or combination of vertex and edge called a *leg*) to generate larger subgraphs. All take a database of graphs as input. GASTON is particularly notable because it can find frequent paths and free trees in addition to general subgraphs.

Some complete algorithms use pattern extension instead of join operations to generate candidate subgraphs. In this method, the new candidate graph is formed by extending a vertex or edge. Examples of these non-Apriori methods include gSpan (Yan et al., 2002) and its variants, CloseGraph (Yan et al., 2003) LCGMiner (Xu & Lei, 2004), ADI-Mine (Wang et al., 2004), and GraphMiner (Wang et al., 2005). Basically, these algorithms build spanning trees for each

graph, using a depth-first search, and order the trees lexicographically into a set of DFS codes. This process results in a hierarchical ordered search tree, over which the algorithm performs a depth first search for subgraph isomorphisms.

FFSM (Huan et al., 2003) is a non-Apriori-like algorithm that allows either the extension or join operations, attempting to increase the efficiency of candidate generation. SPIN (Huan et al., 2004) builds on FFSM, mining maximal frequent subgraphs by finding frequent trees in the graph, and constructing frequent subgraphs from the trees. The algorithm classifies the trees by using a canonical spanning tree as the template for an equivalence class. Pruning mechanisms remove all but the maximally frequent subgraphs. SPIN was designed for applications such as chemical compound analysis, in which frequent patterns are usually tree structures.

MoFa (Borgelt & Berthold, 2002) is a hybrid method that finds connected molecular fragments in a chemical compound database. In this algorithm, atoms are represented as vertices and bonds as edges. MoFa is based on the association mining algorithm *Eclat* rather than Apriori; this algorithm performs a depth first search, and extends the substructure by either a bond or an atom at each iteration.

Heuristic methods include the classic algorithm SUBDUE (Cook & Holder, 1994; Chakravarthy, et al., 2004), a system that uses minimum description length (MDL) to find common patterns that efficiently describe the graph. SUBDUE starts with unique vertex labels and incrementally builds substructures, calculating the MDL heuristic to determine if a substructure is a candidate for compression, and constraining the search space with a beam search. Subgraphs that efficiently compress the graph are replaced with a single vertex, and the discovery process is repeated. This procedure generates a hierarchy of substructures within the complete graph, and is capable of compressing the graph to a single vertex.

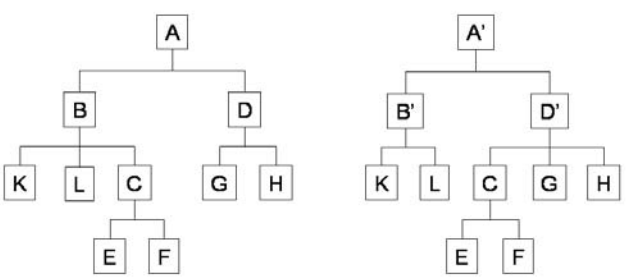
Meinl and Berthold (2004) combined MoFa and the Apriori-based approach FSG (Kuramochi & Karypis, 2003) to mine molecular fragments. The authors exploit the characteristics of the two algorithms; MoFa requires large amounts of memory in the beginning stages, while FSG's memory usage increases in the latter stages of the mining task. The search process begins with FSG and seeds MoFa with frequent subgraphs that reach a specified size. This approach requires a subgraph isomorphism test on new candidates to avoid reusing seeds.

The major mining task for tree-structured data is also frequent subgraph mining, using the same basic four steps as for general graphs. However, unlike the general graph case, polynomial-time subgraph isomorphism algorithms are possible for trees (Horvath et al., 2005; Valiente, 2002). Domains that benefit from frequent subtree mining include web usage (Cooley et al., 1997), bioinformatics (Zhang & Wang, 2006), XML document mining (Zaki & Aggarwal, 2003), and network routing (Cui et al., 2003).

Clustering Methods

Clustering methods find groups of similar subgraphs in a single graph, or similar graphs in a database of several graphs. These algorithms require a similarity

Figure 2. Two bills of materials



or distance table to determine cluster membership. Distance values can be calculated using Euclidean distance, the inner product of graph vectors, unit cost edit distance, or single-level subtree matching. Unit cost edit distance sums the number of unit cost operations needed to insert, delete, or substitute vertices, or change labels until two trees are isomorphic. This measure views the graph as a collection of paths and can also be applied to ordered or unordered trees. Single-level subtree matching (Romanowski & Nagi, 2005) decomposes a tree into subtrees consisting of a parent vertex and its immediate children. In Figure 2, single level subtrees are those rooted at B, D, C for the leftmost tree, and B', C, and D' for the rightmost tree. The algorithm finds the least cost weighted bipartite matchings between the subtrees and sums them to find the total distance between trees A and A'.

This distance measure can be used to cluster tree-based data in any domain where the path-based representation is not applicable, and where otherwise similar trees may have different topologies. Examples include documents, XML files, bills of material, or web pages.

Examples of graph or spatial clustering algorithms are CLARANS (Ng & Han, 1994), ANF (Palmer et al., 2002) and extensions of SUBDUE (Jonyer et al., 2000). Extensive work has been done on clustering in social networks; for an extensive survey, see (Chakrabarti & Faloutsos, 2006).

FUTURE TRENDS

Graph mining research continues to focus on reducing the search space for candidate subgraph generation, and on developing hybrid methods that exploit efficient operations at each stage of the frequent subgraph search. Approximate graph mining is another growing area of inquiry (Zhang et al., 2007).

Web and hypertext mining are also new areas of focus, using mainly visualization methods (e.g., Chen et al., 2004; Yousefi et al., 2004). Other interesting and related areas are link mining (Getoor, 2003), viral marketing (Richardson & Domingos, 2002), and multi-relational data mining (MRDM), which is mining from more than one data table at a time (Domingos, 2003; Dzeroski & DeRaedt, 2002, 2003; Holder & Cook, 2003; Pei et al, 2004).

Graph theory methods will increasingly be applied to non-graph data types, such as categorical datasets (Zaki et al., 2005), object oriented design systems (Zhang et al., 2004), and data matrices (Kuusik et al., 2004). Since tree isomorphism testing appears to be faster and more tractable than general graph isomorphism testing, researchers are transforming general graphs into tree structures to exploit the gain in computational efficiency.

Subgraph frequency calculations and graph matching methods, such as the tree similarity measure, will continue to be investigated and refined to reflect the needs of graph-based data (Vanetik et al., 2006; Romanowski et al., 2006). An additional area of study is how graphs change over time (Faloutsos et al., 2007).

CONCLUSION

Graph research has proliferated as more and more applications for graphically-based data are recognized. However, few current algorithms have been compared using identical datasets and test platforms (Wörlein et al., 2005). This omission makes it difficult to quantitatively evaluate algorithmic approaches to efficient subgraph isomorphism detection, candidate generation, and duplicate subgraph pruning.

Graph mining is an attractive and promising methodology for analyzing and learning from graphical data. The advantages of data mining—the ability to find patterns human eyes might miss, rapidly search through large amounts of data, discover new and unexpected knowledge—can generate major contributions to many different communities.

REFERENCES

Agrawal, R. & Srikant, R. (1994). Fast Algorithms for Mining Association Rules. In *Proceedings of the 20th International Conference on Very Large Databases*, (pp. 487-499). San Mateo, California: Morgan Kaufmann Publishers, Inc.

Borgelt, C. & Berthold, M.R. (2002). Mining molecular fragments: finding relevant substructures of molecules. In *Proceedings of the IEEE International Conference on Data Mining*, (pp. 51-58). Piscataway, NJ: IEEE.

Chakrabarti, D. & Faloutsos, C. (2006). Graph mining: Laws, generators, and algorithms. *ACM Computing Surveys*, 38:1-69.

Chakravarthy, S., Beera, R., and Balachandran, R. (2004) DB-Subdue: Database approach to graph mining. In H. Dai, R. Srikant, and C. Zhang, editors, *Advances in Knowledge Discovery and Data Mining, Proceedings of the 8th PAKDD Conference*, (pp. 341-350). New York: Springer.

Chen, J., Sun, L., Zaïane, O. & Goebel, R. (2004). Visualizing & Discovering Web Navigational Patterns. In *Proceedings of the 7th International Workshop on the Web & Databases*, (pp. 13-18). New York: ACM Press.

Cook, D. & Holder, L. (1994). Substructure Discovery Using Minimum Description Length and Background Knowledge. *Journal of Artificial Intelligence Research*, 1:231-255.

Cooley, R., Mobasher, B. & Srivastava, J. (1997). Web Mining: Information & Pattern Discovery on the World Wide Web. In *Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI97)* (pp. 558-567). Piscataway, NJ: IEEE.

Cui, J., Kim, R., Maggiorini, D., Boussetta, R. & Gerla, M. (2005). Aggregated Multicast - A Comparative Study. *Cluster Computing*, 8(1):15–26.

Domingos, P. (2003). Prospects and Challenges for Multi-Relational Data Mining. *SIGKDD Explorations*, 5: 80-83.

Džeroski, S. & DeRaedt, L. (2002). Multi-Relational Data Mining: A Workshop Report. *SIGKDD Explorations*, 4:122-124.

Džeroski, S. & DeRaedt, L. (2003). Multi-Relational Data Mining: The Current Frontiers. *SIGKDD Explorations*, 5: 100-101.

Faloutsos, C., Kolda, T. & Sun, J. (2007). Mining Large Graphs and Streams Using Matrix and Tensor Tools. In *Proceedings of the 2007 ACM SIGMOD International Conference on Management of data*, (pp. 1174-1174). New York: ACM Press.

Getoor, L. (2003). Link Mining: A New Data Mining Challenge. *SIGKDD Explorations*, 5:84-89.

Goethals, B., Hoekx, E. & Van den Bussche, J. (2005). Mining Tree Queries in a Graph. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 61-69). New York: ACM Press.

Holder, L. & Cook, D. (2003). Graph-Based Relational Learning: Current & Future Directions. *SIGKDD Explorations*, 5:90-93.

Horvath, T., Gartner, T. & Wrobel, S. (2004). Cyclic Pattern Kernels for Predictive Graph Mining. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 158-167). New York: ACM Press.

Huan, J., Wang, W. & Prins, J. (2003). Efficient mining of frequent subgraphs in the presence of isomorphism. In *Proceedings of the 3rd IEEE International Conference on Data Mining*, (pp. 549-552). Piscataway, NJ: IEEE.

Huan, J., Wang, W., Prins, J. & Yang, J. (2004). SPIN: Mining Maximal Frequent Subgraphs from Graph Databases. In *Proceedings of the 10th ACM SIGKDD International Conference Knowledge Discovery & Data Mining*, (pp. 581-586). New York: ACM Press.

Inokuchi, A., Washio, T. & Motoda, H. (2000). An Apriori-based Algorithm for Mining Frequent Substructures from Graph Data. In *Proceedings of the Principles of Data Mining and Knowledge Discovery, 4th European Conference (PKDD 2000)*, (pp. 13-24). Heidelberg: Springer Berlin.

Inokuchi, A. (2004). Mining Generalized Substructures from a Set of Labeled Graphs. In *Proceedings of the 4th IEEE International Conference on Data Mining* (*ICDM 2004*), (pp. 415-418). Piscataway, NJ: IEEE.

Inokuchi, A., Washio, T. & Motoda, H. (2003). Complete Mining of Frequent Patterns from Graphs: Mining Graph Data. *Machine Learning*, 50:321-354.

Jonyer, I., Holder, L. & Cook, D. (2000). Graph-based hierarchical conceptual clustering. *Journal of Machine Learning Research*, 2:19-43.

Kuramochi, M. & Karypis, G. (2005). Finding Frequent Patterns in a Large Sparse Graph. *Data Mining and Knowledge Discovery*, 11(3):243-271.

Kuramochi, M. & Karypis, G. (2004). An Efficient Algorithm for Discovering Frequent Subgraphs. *IEEE Transactions on Knowledge & Data Engineering*, 16:1038-1051.

Kuusik, R., Lind, G. & Võhandru, L. (2004). Data Mining: Pattern Mining as a Clique Extracting Task. In *Proceedings of the 6th International Conference on Enterprise Information Systems (ICEIS 2004)* (pp. 519-522). Portugal: Institute for Systems and Technologies of Information, Control and Communication. (INSTICC)

Meinl, T. & Berthold, M. (2004). Hybrid fragment mining with MoFa and FSG. In *Proceedings of the IEEE International Conference on Systems, Man & Cybernetics* (pp. 4559-4564). Piscataway, NJ: IEEE.

Ng, R. & Han, J., (1994). Efficient & Effective Clustering Methods for Spatial Data Mining. In *Proceedings of the 20th International Conference on Very Large DataBases*, (pp 144-155). New York: ACM Press.

Nijssen, S. & Kok, J. (2004). Frequent Graph Mining and Its Application to Molecular Databases. In *Proceedings of the IEEE International Conference on Systems, Man & Cybernetics*, (pp. 4571-4577). Piscataway, NJ: IEEE.

Palmer, C., Gibbons, P., & Faloutsos, C. (2002). ANF: A Fast and Scalable Tool for Data Mining in Massive Graphs. In *Proceedings of the 8th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, (pp. 81-90). New York: ACM Press.

Pei, J., Jiang, D. & Zhang, A. (2004). On Mining Cross-Graph Quasi-Cliques. In *Proceedings of the 10th ACM SIGKDD International Conference Knowledge Discovery and Data Mining*, (pp. 228-238). New York: ACM Press.

Richardson, M. & Domingos, P. (2002). Mining Knowledge-Sharing Sites for Viral Marketing. In *Proceedings of the 8th ACM SIGKDD International Conference Knowledge Discovery & Data Mining*, (pp. 61-70). New York: ACM Press.

Romanowski, C.J. & Nagi, R. (2005). On clustering bills of materials: A similarity/distance measure for unordered trees. *IEEE Transactions on Systems, Man & Cybernetics, Part A: Systems & Humans*, 35: 249-260.

Romanowski, C.J., Nagi, R. & Sudit, M., (2006). Data mining in an engineering design environment: OR applications from graph matching. *Computers & Operations Research*, 33(11): 3150-3160.

Valiente, G., 2002. *Algorithms on Trees and Graphs.* Berlin: Springer-Verlag, Inc.

Vanetik, N. & Gudes, E. (2004). Mining Frequent Labeled & Partially Labeled Graph Patterns. In *Proceedings of the International Conference on Data Engineering (ICDE (2004),* (pp.91-102). Piscataway, NJ: IEEE Computer Society.

Vanetik, N., Gudes, E. & Shimony, S.E. (2006). Support measures for graph data. *Data Mining & Knowledge Discovery*, 13:243-260.

Wang, C., Wang, W., Pei, J., Zhu, Y. & Shi, B. (2004). Scalable Mining of Large Disk-based Graph Databases. In *Proceedings of the 10th ACM SIGKDD International Conference Knowledge Discovery and Data Mining*, (pp. 316-325). New York: ACM Press.

Wang, W., Wang, C., Zhu, Y., Shi, B., Pei, J. Yan, X. & Han, J. (2005). GraphMiner: A Structural Pattern-Mining System for Large Disk-based Graph Databases and Its Applications. In *Proceedings of the ACM SIGMOD*, (pp.879-881). New York: ACM Press.

Washio, T. & Motoda, H. (2003). State of the Art of Graph-based Data Mining. *SIGKDD Explorations*, 5:59-68.

Wörlein, M., Meinl, T., Fischer, I. & Philippsen, M. (2005). A Quantitative Comparison of the Subgraph Miners MoFa, gSpan, FFSM, and Gaston. *Principles of Knowledge Discovery & Data Mining (PKDD 2005)*, (pp. 392-403). Heidelberg: Springer-Verlag.

Xu, A & Lei, H. (2004). LCGMiner: Levelwise Closed Graph Pattern Mining from Large Databases. In *Proceedings of the 16th International Conference on Scientific & Statistical Database Management*, (pp. 421-422). Piscataway, NJ: IEEE Computer Society.

Yan, X & Han, J. (2002). gSpan: Graph-based Substructure Pattern Mining. In *Proceedings of the IEEE International Conference on Data Mining (ICDM 2002)*, (pp. 721-724). Piscataway, NJ: IEEE.

Yan, X. & Han, J. (2003). CloseGraph: Mining Closed Frequent Graph Patterns. In *Proceedings of the 9th ACM*

SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '03), (pp. 286-295). New York: ACM Press.

Yousefi, A., Duke, D. & Zaki, M. (2004). Visual Web Mining. In *Proceedings of the 13th International World Wide Web Conference*, (pp. 394-395). New York: ACM Press.

Zaki, M. & Aggarwal, C. (2003). XRules: an effective structural classifier for XML data. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '03)*, (pp. 316–325). New York: ACM Press.

Zaki, M., Peters, M., Assent, I. & Seidl, T. (2005). CLICKS: An Effective Algorithm for Mining Subspace Clusters in Categorical Datasets. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '05)*, (pp. 736-742). New York: ACM Press.

Zhang, S. & Wang, J. T-L. (2006). Mining Frequent Agreement Subtrees in Phylogenetic Databases. In *Proceedings of the 6th SIAM International Conference on Data Mining (SDM 2006)*, (pp. 222–233). Philadelphia: SIAM Press.

Zhang, S., Yang, J. & Cheedella, V. (2007). Monkey: Approximate Graph Mining Based on Spanning Trees. In *Proceedings of IEEE 23rd International Conference on Data Engineering*, (pp. 1247-1249). Piscataway, NJ: IEEE.

Zhang, Z., Li, Q. & Ben, K. (2004). A New Method for Design Pattern Mining. In *Proceedings of the 3rd International Conference on Machine Learning & Cybernetics*, (pp. 1755-1759). New York: IEEE.

KEY TERMS

Acyclic Graph: A graph that does not contain a cycle, or a path from one vertex that ends up back at the same vertex.

Directed Graph: A graph in which pairs of vertices are connected with edges that have a single direction. Hence, if Vertex 1 is connected to Vertex 2 by a directed edge, then Vertex 1 is considered the parent of Vertex 2 and cannot be reached from Vertex 2 via that edge.

Lexicographic Order: Also called "dictionary order"; subgraphs can have many isomorphs, and ordering them allows the algorithm to choose the minimum lexicographic subgraph to represent all of a particular subgraph's isomorphs. Thus, the search space is reduced since an algorithm will not need to test multiple copies of the same subgraph.

Supergraph: A graph that contains one or more subgraphs. If subgraph *G'* is contained in graph *G*, then *G* is a supergraph of *G'*.

Support: In frequent subgraph mining, *support* is the number of times a particular subgraph pattern appears in a database of graphs. Users designate a minimum support value; subgraphs that occur fewer times than the minimum support are infrequent, and would not be used for further candidate subgraph generation.

Topology: The number and arrangement of vertices and edges in a graph.

Weighted Bipartite Matching: A method of pairwise matching of two sets of nodes or vertices, where edges between them carry weights. An example is the marriage problem, where one set represents the women and one set represents the men; the edges between them represent how much each person is attracted to the other, and the goal is to generate a set of pairs with the highest possible sum of the matched weights.

Guide Manifold Alignment by Relative Comparisons

Liang Xiong
Tsinghua University, China

Fei Wang
Tsinghua University, China

Changshui Zhang
Tsinghua University, China

INTRODUCTION

When we are faced with data, one common task is to learn the correspondence relationship between different data sets. More concretely, by learning data correspondence, samples that share similar intrinsic parameters, which are often hard to estimate directly, can be discovered. For example, given some face image data, an alignment algorithm is able to find images of two different persons with similar poses or expressions. We call this technique the alignment of data. Besides its usage in data analysis and visualization, this problem also has wide potential applications in various fields. For instance, in *facial expression recognition*, one may have a set of standard labeled images with known expressions, such as *happiness*, *sadness*, *surprise*, *anger and fear*, of a particular person. Then we can recognize the expressions of another person just by aligning his/her facial images to the standard image set. Its application can also be found directly in pose estimation. One can refer to (Ham, Lee & Saul, 2005) for more details.

Although intuitive, without any premise this alignment problem can be very difficult. Usually, the samples are distributed in high-dimensional observation spaces, and the relation between features and samples' intrinsic parameters can be too complex to be modeled explicitly. Therefore, some hypotheses about the data distribution are made. In the recent years, the manifold assumption of data distribution has been very popular in the field of data mining and machine learning. Researchers have realized that in many applications the samples of interest are actually confined to particular subspaces embedded in the high-dimensional feature space (Seung & Lee, 2000; Roweis & Saul, 2000). Intuitively, the manifold assumption means that certain groups of samples are lying in a non-linear low-dimensional subspace embedded in the observation space. This assumption has been verified to play an important role in human perception (Seung & Lee, 2000), and many effective algorithms are developed under it in the recent years. Under the manifold assumption, structural information of data can be utilized to facilitate the alignment. Figure 1 illustrates two 1-D manifolds embedded in a 2-D plane.

Besides, since we do not know the relationship between observations and the underlying parameters, additional supervisions are needed to guide the aligning process. Usually, we will require information about a subset of samples and the structure of the whole data set in order to infer about the rest of samples. Thus,

Figure 1. Two 1-D manifolds embedded in a 2-D plane

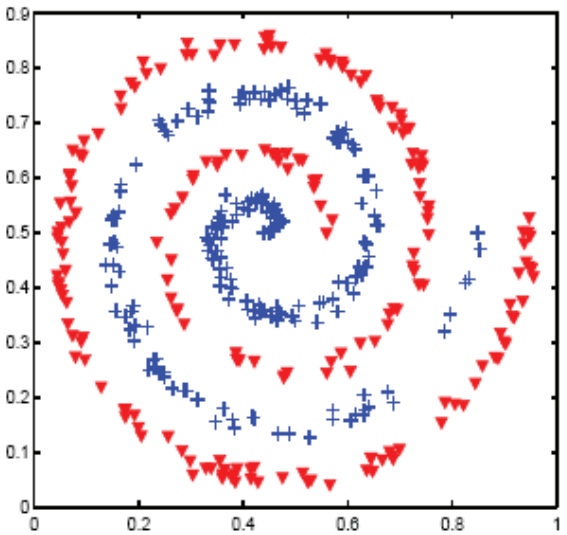
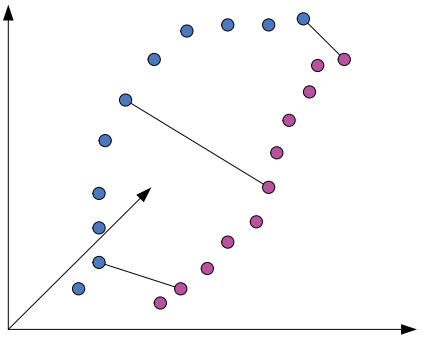


Figure 2. Guide the alignment of two manifolds by pairwise correspondence. A black line represents the supervision which indicates two samples with the same underlying parameters.



the alignment is often done in a semi-supervised way. The most common supervision used is pairwise correspondence, which specifies two samples that share the same parameters as shown in figure 2.

Take the facial expression alignment for example. In this task we are given face images of two different persons A and B, and the problem is to find two images, one of which is from A and the other is from B, with similar facial expression. Template matching in the feature space is not feasible here since different faces may have very different appearances. At present, directly estimating the expression parameters is also difficult because of the limitation of our knowledge and the variability of data. So we assume that images from the same person are lying on a low-dimensional manifold. Now the problem seems easier because we are now dealing with two structures, such as two curves, instead of discrete points. However, we still do not know how these two structures should be aligned. Then supervisions are needed to tell us how they correspond.

There have already been several methods proposed to align manifolds in a semi-supervised way (Ham, Ahn & Lee, 2006; Ham, Lee & Saul, 2003; Ham, Lee & Saul, 2005; Verbeek, Roweis & Vlassis, 2004; Verbeek & Vlassis, 2006). Specifically, they usually assumed that some pair-wise correspondences of samples in different data sets were already known, and then this information would be used to guide the alignment. However, in practice it might be difficult to obtain and use such information since:

1. The sizes of data sets can be very large, then finding high-quality correspondences between them can be very time consuming and even intractable.
2. There may be ambiguities in the images, which makes explicit matching a hard task. Brutally determine and enforce these unreliable constraints may lead to poor results;
3. Sometimes it may be hard to find the exact correspondences when the available samples are scarce. This situation may happen when the data source is restricted and users are only allowed to access a small subset.

To solve the above problems, we could apply another type of supervision to guide the process of manifold alignment. In particular, we consider a relative and qualitative supervision of the form "A is closer to B than A is to C". We believe that this type of information is more easily available in practice than traditional correspondence-based information. With the help of such information, we show that the manifold alignment problem can be formulated as a *Quadratically Constrained Quadratic Programming* (QCQP) (Boyd & Vandenberghe, 2004) problem. To make the optimization tractable, we further relax it to a *Semi-Definite Programming* (SDP) (Boyd & Vandenberghe, 2004) problem, which can be readily solved by popular optimization software packages. Besides, under this formulation we are able to incorporate both relative relations and correspondences to align manifolds in a very flexible way.

BACKGROUND

Before alignment, we need an infrastructure to capture the structure of each manifold. First, we will introduce manifold embedding.

The embedding algorithms find coordinates in a novel low-dimensional space for the data points so that certain geometric relations between them in the feature space are retained in the embedded space. Here we complete this task by graph regularization as in *Laplacian Eigenmaps* (Belkin & Niyogi, 2003). An undirected weighted graph is constructed over the data points to characterize their relations. Then this method aims at preserving the connectivity in this graph *i.e.* if two points are originally connected, then after embedding they are also connected; if two points

are close to each other in the feature space, then they will still be close in the embedded space.

The above problem is solved by using the Laplacian of graphs. More concretely, for data set $X = \{x_1, x_2, \cdots, x_N\} \subset R^{D_X}$, where D_X is the dimensionality of samples in X, we can construct a graph $G_X = (V_X, E_X)$, where $V_X = X$ corresponds to the nodes of G_X and E_X is the edge set in the graph. There is a non-negative weight w_{ij} associated with each edge $e_{ij} \in E_X$. When constructing the graph, we want to only capture local relations between data according to the traits of manifold. So k-nearest neighbor graphs are often used, in which only nearby nodes are connected. Plus, the edge weight w_{ij} are used to represent the similarity between nodes. If x_i and x_j are close to each other then w_{ij} is large, and vice versa. As for the calculation of edge weight w_{ij}, several methods have been proposed. Popular choices include the Gaussian kernel (Zhu, 2005) and LLE reconstruction (Roweis. & Saul, 2000; Wang & Zhang, 2006). Further discussion on this issue is beyond the scope of this chapter. Readers can refer to (Zhu, 2005) for more details. Here we assume that the edge weight is symmetric *i.e.* $w_{ij} = w_{ji}$, which means that the similarity from x_i to x_j is the same as that from x_j to x_i.

We collect all these edge weights to form an $N \times N$ weight matrix $\mathbf{W}_X$ with its entry $\mathbf{W}_X(i, j) = \mathbf{W}_{ij}$. The degree matrix $\mathbf{D}_X$ is an $N \times N$ diagonal matrix with its i-th diagonal entry $\mathbf{D}_X(i,i) = \sum_j w_{ij}$. Given $\mathbf{W}_X$ and $\mathbf{D}_X$, the combinatorial graph Laplacian is defined as

$$\mathbf{L}_X = \mathbf{D}_X - \mathbf{W}_X \tag{1}$$

Generally, the graph Laplacian can be used to enforce the smoothness constraint on a graph i.e. a manifold (Belkin & Niyogi, 2003), which means that the properties of nearby nodes should also be close. Suppose that there is a property function $p_i = (x_i)$ associated with each node, then the smoothness of p on the graph can be achieved by solving the following problem

$$p = \min_p S = \min_p p\mathbf{L}_X p^T \tag{2}$$

where $p = [p_1, p_2, \cdots, p_N]$. To see this, just consider the fact that

$$S = p\mathbf{L}_X p^T = \frac{1}{2}\sum_{i,j} w_{ij}(p_i - p_j)^2 \tag{3}$$

The same idea can be used to calculate the embedding of data by considering the embedded coordinates as the property function p in (2). Specifically, let $\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, \cdots \mathbf{f}_N] \subset R^{d \times N}$ be the embedded coordinates of X, then $\mathbf{F}$ can be obtained by

$$\min_{\mathbf{F}} tr(\mathbf{F}\mathbf{L}_X\mathbf{F}^T) \tag{4}$$

where $tr(\cdot)$ means the trace of a matrix. By optimization (4) along with proper scale and translation invariance constraints, we can get a low-dimensional representation of X that preserves the underlying manifold structure.

MAIN FOCUS

Co-Embedding and Traditional Alignment

In the previous section, we showed how to calculate the embedding of a manifold. In essence, manifold alignment can also be seen as the embedding of manifolds. The difference is that in alignment, multiple manifolds have to be handled and their correspondence should be taken into consideration. Take the alignment of two manifolds X and Y for example, if we do not take account of correspondence, the embedding of these two manifolds can be obtained by *co-embedding*

$$\min_{\mathbf{F},\mathbf{G}} \left(tr(\mathbf{F}\mathbf{L}_X\mathbf{F}^T) + tr(\mathbf{G}\mathbf{L}_Y\mathbf{G}^T)\right)$$
$$s.t. \begin{cases} tr(\mathbf{F}\mathbf{F}^T) = tr(\mathbf{G}\mathbf{G}^T) = 1 \\ \mathbf{F}\mathbf{e} = 0, \mathbf{G}\mathbf{e} = 0 \end{cases} \tag{5}$$

where $\mathbf{F}$ and $\mathbf{G}$ are the embedded coordinates of X and Y respectively. The constraints are for scale and translation invariance, and $\mathbf{e}$ is an all-one column vector. These two constraints apply to all the optimization problems in the rest of this article, but for notational simplicity we will not write them explicit.

To achieve alignment, extra constraints or objective terms have to be appended to the co-embedding in (5). Traditional methods use pair-wise correspondence constraints to guide the alignment. Usually, a *fitting term* is added to the objective as (Ham, Lee & Saul, 2003; Ham, Lee & Saul, 2005; Verbeek, Roweis & Vlassis, 2004; Verbeek & Vlassis, 2006)

$$\min_{\mathbf{F},\mathbf{G}} \mu \sum_{i=1}^{c} \left\| \mathbf{f}_i - \mathbf{g}_i \right\| + tr(\mathbf{F}\mathbf{L}_X\mathbf{F}^T) + tr(\mathbf{G}\mathbf{L}_Y\mathbf{G}^T) \quad (6)$$

where μ is the parameter to control the tradeoff between the embedding smoothness and the match precision, while c is the number of given corresponding pairs. This formulation ensures that two indicated samples are close to each other in the embedded space.

Alignment Guided by Relative Comparison – A Quadratic Formulation

As we state in the introduction, pair-wise supervision for manifold alignment has some drawbacks for large data sets or when correspondence is not well defined. So we instead turned to use another type of guidance: relative comparison. Here, this guidance takes the form of "y_i is closer to x_j than x_k", notated by an ordered 3-tuple $t_c = \{y_i, x_j, x_k\}$. We translate t_c into the requirement that in the embedded space the distance between $\mathbf{g}_i$ and $\mathbf{f}_j$ is smaller than that between $\mathbf{g}_i$ and $\mathbf{f}_k$. More formally, t_c leads to the constraint

$$\left\| \mathbf{g}_i - \mathbf{f}_j \right\|^2 \le \left\| \mathbf{g}_i - \mathbf{f}_k \right\|^2 \quad (7)$$

Let $T = \{t_c\}^C_{c=1}$ denote the constraint set. Under these relative comparison constraints, our optimization problem can be formulated as

$$\min_{\mathbf{F},\mathbf{G}} tr(\mathbf{F}\mathbf{L}_X\mathbf{F}^T) + tr(\mathbf{G}\mathbf{L}_Y\mathbf{G}^T)$$
$$s.t. \begin{cases} \forall \{y_i, x_j, x_k\} \in T, \left\| \mathbf{g}_i - \mathbf{f}_j \right\|^2 \le \left\| \mathbf{g}_i - \mathbf{f}_k \right\|^2 \end{cases} \quad (8)$$

Let $\mathbf{H} = [\mathbf{F}, \mathbf{G}]$, $\mathbf{L} = \begin{bmatrix} \mathbf{L}_X & \\ & \mathbf{L}_Y \end{bmatrix}$, we can further simplify the problem to

$$\min_{\mathbf{H}} tr(\mathbf{H}\mathbf{L}\mathbf{H}^T)$$
$$s.t. \begin{cases} \forall \{y_i, x_j, x_k\} \in T, \left\| \mathbf{h}_{i+N} - \mathbf{h}_j \right\|^2 \le \left\| \mathbf{h}_{i+N} - \mathbf{h}_k \right\|^2 \end{cases} \quad (9)$$

Now we have formulated our tuple-constrained optimization as a *Quadratically Constrained Quadratic Programming* (QCQP) (Boyd & Vandenberghe, 2004) problem. However, since the relative comparison constraints are not *convex*, then 1) the solution is computationally difficult to derive and 2) the solution can be trapped in local minima. Therefore, a re-formulation is needed to make it tractable.

Alignment Guided by Relative Comparison – A Semi-definite Formulation

In this section we show how to convert the QCQP problem in the pervious section to a *Semi-Definite Programming* (SDP) (Boyd & Vandenberghe, 2004) problem. This conversion is motivated by the quadratic form of (9) and its relationship to the kernel matrix of data. The kernel matrix is defined as $\mathbf{K} = \mathbf{H}^T\mathbf{H}$ with its element $\mathbf{K}(i, j) = \mathbf{h}_i^T \mathbf{h}_j$. By some simple algebraic manipulations, problem (9) can be fully formulated in terms of $\mathbf{K}$. For notational simplicity, we divided $\mathbf{K}$ into four blocks as:

$$\mathbf{K} = \begin{bmatrix} \mathbf{F}^T\mathbf{F} & \mathbf{F}^T\mathbf{G} \\ \mathbf{G}^T\mathbf{F} & \mathbf{G}^T\mathbf{G} \end{bmatrix} = \begin{bmatrix} \mathbf{K}^{FF} & \mathbf{K}^{FG} \\ \mathbf{K}^{GF} & \mathbf{K}^{GG} \end{bmatrix}.$$

The objective function is

$$tr(\mathbf{H}\mathbf{L}\mathbf{H}^T) = tr(\mathbf{L}\mathbf{H}^T\mathbf{H}) = tr(\mathbf{L}\mathbf{K}) \quad (10)$$

The relative comparison constraints are

$$\begin{aligned} &\left\| \mathbf{h}_{i+N} - \mathbf{h}_j \right\|^2 \le \left\| \mathbf{h}_{i+N} - \mathbf{h}_k \right\|^2 \\ &\Leftrightarrow -2\mathbf{h}_{i+N}^T\mathbf{h}_j + 2\mathbf{h}_{i+N}^T\mathbf{h}_k + \mathbf{h}_j^T\mathbf{h}_j - \mathbf{h}_k^T\mathbf{h}_k \le 0 \\ &\Leftrightarrow -2\mathbf{K}_{i+N,j} + 2\mathbf{K}_{i+N,k} + \mathbf{K}_{j,j} - \mathbf{K}_{k,k} \le 0 \end{aligned} \quad (11)$$

The scale invariance constraints are

$$\begin{aligned} &tr(\mathbf{F}\mathbf{F}^T) = tr(\mathbf{K}^{FF}) = 1 \\ &tr(\mathbf{G}\mathbf{G}^T) = tr(\mathbf{K}^{GG}) = 1 \end{aligned} \quad (12)$$

The translation invariance is achieved by constraints

$$\sum_{i,j} K_{i,j}^{FF} = 0, \sum_{i,j} K_{i,j}^{GG} = 0 \quad (13)$$

Finally, to be a valid kernel matrix, $\mathbf{K}$ must be positive semi-definite, resulting that

$$\mathbf{K} \succeq 0 \quad (14)$$

Combining (10) to (14), the new optimization problem is derived as

$$\min_{\mathbf{K}} tr(\mathbf{LK})$$
$$s.t.\begin{cases}\forall\{y_i,x_j,x_k\}\in T,-2\mathbf{K}_{i+N,j}+2\mathbf{K}_{i+N,k}+\mathbf{K}_{j,j}-\mathbf{K}_{k,k}\le 0\\ \mathbf{K}\succeq 0\end{cases}$$

Moreover, to avoid the case of empty feasible set and to encourage the influence of the guidance information, we introduce slack variables $E = \{\varepsilon_c\}^C_{c=1}$ and relax the problem as follows

$$\min_{\mathbf{K},E} tr(\mathbf{LK})+\alpha\sum_{c=1}^{C}\varepsilon_c$$
$$s.t.\begin{cases}\forall\{y_i,x_j,x_k\}\in T,-2\mathbf{K}_{i+N,j}+2\mathbf{K}_{i+N,k}+\mathbf{K}_{j,j}-\mathbf{K}_{k,k}\le 0\\ \mathbf{K}\succeq 0\\ \forall\varepsilon_c\in E,\varepsilon_c\le 0\end{cases}$$
(11)

where α is a parameter to balance the data's inherent structure and the supervised information. When α is small, the embedding is dominated by the manifold structure; otherwise, the prior knowledge (i.e. the supervision plays a more important role).

Problem (16) is a standard SDP problem which can be readily solved by optimization software packages such as SeDuMi (Sturm, 1999). Therefore, the above aligning algorithm is called Semi-Definite Manifold Alignment (SDMA). When the kernel matrix **K** is solved, the embedded coordinates **F** and **G** can be recovered from **K**'s dominant eigenvectors as in kernel PCA (Schölkopf, Smola & Müller, 1998). Then, corresponding sample pairs can be obtained by either finding nearest neighbor or minimizing the total matching cost. More details about this method can be found in (Xiong, Wang & Zhang, 2007). An experimental result of aligning head poses by SDMA is illustrated in figure 3.

FUTURE TRENDS

One common footstone of manifold learning methods is the characterization of data's underlying manifold structure, which is the focus of manifold embedding algorithms. This is also true for manifold alignment. The more accurate manifolds are captured, the better the alignment will be. Thus alignment algorithms should be updated with the emergence of new embedding methods. In SDMA, we choose to use the graph regularization method. Recently, (Weinberger & Saul, 2006; Weinberger, Sha & Saul, 2004) proposed to achieve this goal by preserving the local isometry of data. This new infrastructure can also be integrated with SDMA to improve its performance.

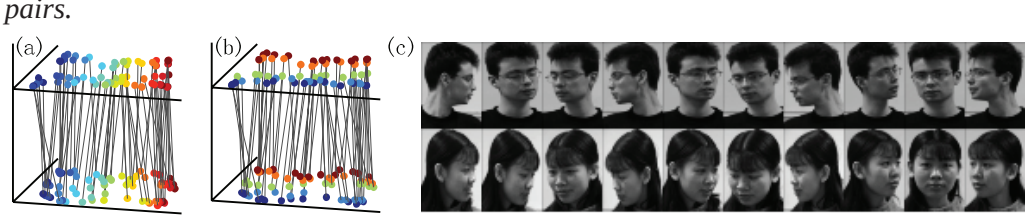
Another important aspect of manifold alignment is how to guide the process of finding correspondences. Traditional pair-wise constraints may not be suitable for some large data sets. SDMA provides a more flexible alternative. In the future, more forms of supervision should be developed to achieve superior performance and wider applicability.

To be a practical utility for data mining, the most important improvement needed for SDMA and other alignment algorithms are their speed and storage efficiencies. Currently, these methods rely on standard optimization routines and are often not directly applicable to large data sets. In general, this issue will benefit from the development of more efficient numeric optimization algorithms and careful algorithm design with respect to specific data sets.

CONCLUSION

Finding correspondences between different data set is an intuitive, useful, yet difficult problem. However, it

Figure 3. Alignment of head pose images. (a) and (b) shows the embedding by SDMA and the correspondences found. Points are colored according to horizontal angles in (a) and vertical angles in (b). (c) shows some samples of matched pairs.



can be made tractable by utilizing the manifold structure of data. Further, some human supervision can be incorporated to guide how two manifolds can be aligned. These two steps consists of the main component of a align algorithm. To achieve maximum flexibility, *Semi-Definite Manifold Alignment* (SDMA) chooses to guide the alignment by relative comparisons in the form of "A is closer to B than A is to C", in contrast to traditional methods that uses pairwise correspondences as supervision. This new type of constraint is well defined on many data sets and very easy to obtain. Therefore, its potential usage is very wide.

REFERENCES

Belkin, M. & Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation, *Neural Computation, 15*, 1373-1396.

Boyd, S.P. & Vandenberghe, L. (2004). *Convex Optimization*, Cambridge, UK.

Ham, J., Ahn, I. & Lee, D. (2006). Learning a manifold-constrained map between image sets: Applications to matching and pose estimation, In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition.*

Ham, J., Lee, D. & Saul, L. (2003). Learning high dimensional correspondence from low dimensional manifolds, workshop on the continuum from labeled to unlabeled data in machine learning and data mining, In: *Proceedings of International Conference on Machine Learning.*

Ham, J., Lee, D. & Saul, L. (2005). Semi-supervised alignment of manifolds, In: *Proceedings of the 8th International Workshop on Artificial Intelligence and Statistics.*

Roweis, S.T. & Saul, L.K. (2000). Nonlinear Dimensionality Reduction by Locally Linear Embedding, *Science, 290*, 2323-2326.

Schölkopf, B., Smola, A., Müller, K. (1998). Nonlinear component analysis as a kernel eigenvalue problem, *Neural Computation, 10*, 1299-1319.

Seung, H.S. & Lee, D.D. (2000). The Manifold Ways of Perception, *Science, 290*, 2268-2269.

Sturm, J.F. (1999). Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones, *Optimization Methods and Software, 11-12*, 625–653.

Verbeek, J., Roweis, S. & Vlassis, N. (2004). Non-linear CCA and PCA by alignment of local models, In: *Advances in Neural Information Processing Systems 17.*

Verbeek, J. & Vlassis, N. (2006). Gaussian fields for semi-supervised regression and correspondence learning, *Pattern Recognition, 39(10)*, 1864-1875.

Wang, F. & Zhang, C. (2006). Label propagation through linear neighborhoods, In: *Proceedings of International Conference on Machine Learning.*

Weinberger, K.Q. & Saul, L.K. (2006). Unsupervised learning of image manifolds by semidefinite programming, *International Journal of Computer Vision, 70(1)*, 77-90.

Weinberger, K., Sha, F. & Saul, L.K. (2004). Learning a kernel matrix for nonlinear dimensionality reduction, In: *Proceeding of International Conference on Machine Learning.*

Xiong, L., Wang, F. & Zhang, C. (2007). Semi-definite manifold alignment, *European Conference on Machine Learning (ECML), Lecture Notes in Computer Science, 4701*, 773-781, Springer.

Zhu, X. (2005). *Semi-supervised learning with graphs*, PhD thesis, CMU.

KEY TERMS

Convex Optimization: An optimization problem which consists of an objective function, inequality constraints, and equality constraints. The objective and the inequality constraints must be convex, and the equality constraint must be affine. In convex optimization, any locally optimal solution is also globally optimal.

Kernel Matrix: A square matrix that represents all the pairwise similarities within a set of vectors. When the similarity is measured by inner product, the kernel matrix is also the Gram matrix of data.

Manifold: A topological space which is locally Euclidean, i.e., "flat" in a small neighborhood.

Manifold Alignment: A special case of manifold embedding. The difference is that multiple manifolds are handled and in the embedded space correspond points from different manifolds are close to each other.

Manifold Embedding: A technique to represent manifolds in a certain space, which is usually more perceivable to human, while keeping the manifold's geometric or algebraic properties preserved.

Semi-Definite Programming (SDP): An optimization technique that deals with problems over symmetric positive semi-definite matrix. Its objective and the constraints are both linear.

Semi-Supervised Learning: A class of machine learning methods. In contrast to traditional supervised learning which relies solely on labeled data, semi-supervised learning incorporates unlabeled data in the learning process as well.

Guided Sequence Alignment

Abdullah N. Arslan
University of Vermont, USA

INTRODUCTION

Sequence alignment is one of the most fundamental problems in computational biology. Ordinarily, the problem aims to align symbols of given sequences in a way to optimize similarity score. This score is computed using a given scoring matrix that assigns a score to every pair of symbols in an alignment. The expectation is that scoring matrices perform well for alignments of all sequences. However, it has been shown that this is not always true although scoring matrices are derived from known similarities. Biological sequences share common sequence structures that are signatures of common functions, or evolutionary relatedness. The alignment process should be guided by constraining the desired alignments to contain these structures even though this does not always yield optimal scores. Changes in biological sequences occur over the course of millions of years, and in ways, and orders we do not completely know. Sequence alignment has become a dynamic area where new knowledge is acquired, new common structures are extracted from sequences, and these yield more sophisticated alignment methods, which in turn yield more knowledge. This feedback loop is essential for this inherently difficult task.

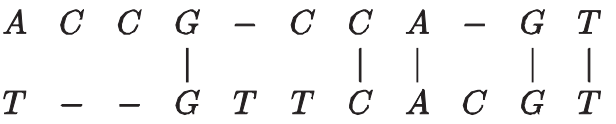
The ordinary definition of sequence alignment does not always reveal biologically accurate similarities. To overcome this, there have been attempts that redefined sequence similarity. Huang (1994) proposed an optimization problem in which close matches are rewarded more favorably than the same number of isolated matches. Zhang, Berman & Miller (1998) proposed an algorithm that finds alignments free of low scoring regions. Arslan, Eğecioğlu, & Pevzner (2001) proposed length-normalized local sequence alignment for which the objective is to find subsequences that yield maximum length-normalized score where the length-normalized score of a given alignment is its score divided by sum of subsequence-lengths involved in the alignment. This can be considered as a context-dependent sequence alignment where a high degree of local similarity defines a context. Arslan, Eğecioğlu, & Pevzner (2001) presented a fractional programming algorithm for the resulting problem. Although these attempts are important, some biologically meaningful alignments can contain *motif*s whose inclusions are not guaranteed in the alignments returned by these methods. Our emphasis in this chapter is on methods that guide sequence alignment by requiring desired alignments to contain given common structures identified in sequences (motifs).

BACKGROUND

Given two strings S_1, and S_2, the *pairwise sequence alignment* can be described as a writing scheme such that we use a two-row-matrix in which the first row is used for the symbols of S_1, and the second row is used for those of S_2, and each symbol of one string can be aligned to (i.e. it appears on the same column with) a symbol of the other string, or the blank symbol ´-´. A matrix obtained this way is called an alignment matrix. No column can be entirely composed of blank symbols. Each column has a weight. The score of an alignment is the total score of the columns in the corresponding alignment matrix. Fig. 1 illustrates an example alignment between two strings ACCGCCAGT and TGTTCACGT.

The following is the Needleman-Wunsch global alignment formulation (Durbin et al., 1998) that, for two given strings $S_1[1] \ldots S_1[n]$ and $S_2[1] \ldots S_2[m]$, computes

Figure 1. An example alignment with five matches



$$H_{i,j} = \max\{ H_{i-1,j}+\gamma(S_1[i],'-'),\ H_{i-1,j-1}+\gamma(S_1[i],S_2[j]),\ H_{i,j-1}+\gamma('-',S_2[j]) \} \quad (1)$$

for all $i, j,\ 1 \le i \le n,\ 1 \le j \le m$, with the boundary values $H_{0,0} = 0$, $H_{0,j} = H_{0,j-1} + \gamma('-',S_2[j])$, and $H_{i,0} = H_{i-1,0} + \gamma(S1[i], '-')$ where γ is a given score function. Then $H_{n,m}$ is the maximum global alignment score between S_1 and S_2. For the strings in Fig. 1, if $\gamma(x,y)$=1 when x=y, and 0 otherwise, then the maximum alignment score *is* $H_{9,9}$=5. The figure shows an optimal alignment matrix with 5 matches each indicated by a vertical line segment. The well-known Smith-Waterman local alignment algorithm (Durbin et al., 1998) modifies Equation (1) by adding 0 as a new max-term. This means that a new local alignment can start at any position in the alignment matrix if a positive score cannot be obtained by local alignments that start before this position. For the example strings in Fig. 1, if the score of a match is +1, and each of all other scores is -1, then the optimum local alignment score is 3, and it is obtained between the suffixes CAGT, and CACGT of the two strings, respectively.

The definition of pairwise sequence alignment for a pair of sequences can be generalized to the *multiple sequence alignment* problem (Durbin et al., 1998). A multiple sequence alignment of k sequences involves a k-row alignment matrix, and there are various scoring schemes (e.g. sum of pairwise distances) assigning a weight to each column.

Similarity measures based on computed scores only does not always reveal biologically relevant similarities (Comet & Henry, 2002). Some important local similarities can be overshadowed by other alignments (Zhang, Berman & Miller, 1998). A biologically meaningful alignment should include a region in it where common sequence structures (if they exist) are aligned together although this would not always yield higher scores. It has also been noted that biologists favor integrating their knowledge about common patterns, or structures into the alignment process to obtain biologically more meaningful similarities (Tang et al., 2002; Comet & Henry, 2002). For example, when comparing two protein sequences it may be important to take into account a common specific or putative structure which can be described as a subsequence. This gave rise to a number of constrained sequence alignment problems. Tang et al. (2002) introduced the *constrained multiple sequence alignment* (CMSA) problem where the constraint for the desired alignment(s) is inclusion of a given subsequence. This problem and its variations have been studied in the literature, and different algorithms have been proposed (e.g. He, Arslan, & Ling, 2006; Chin et al., 2003).

Arslan and Eğecioğlu (2005) suggested that the constraint could be inclusion of a subsequence within a given edit distance (number of edit operations to change one string to another). They presented an algorithm for the resulting constrained problem. This was a step toward allowing in the constraint patterns that may slightly differ in each sequence.

Arslan (2007) introduced the *regular expression constrained sequence alignment* (RECSA) problem in which alignments are required to contain a given common sequence described by a given regular expression, and he presented an algorithm for it.

MAIN FOCUS

Biologists prefer to incorporate their knowledge into the alignment process by guiding alignments to contain known sequence structures. We focus on motifs that are described as a subsequence, or a regular expression, and their use in guiding sequence alignment.

Subsequence Motif

The constraint for the alignments sought can be inclusion of a given pattern string as a subsequence. A motivation for this case comes from the alignment of RNase (a special group of enzymes) sequences. Such sequences are all known to contain "HKH'' as a substring. Therefore, it is natural to expect that in an alignment of RNase sequences, each of the symbols in "HKH'' should be aligned in the same column, i.e. an alignment sought satisfies the constraint described by the sequence "HKH". The alignment shown in Fig. 1 satisfies the constraint for the subsequence pattern "CAGT".

Chin et al. (2003) present an algorithm for the constrained multiple sequence alignment (CMSA) problem. Let S_1, S_2, ..., S_n be given n sequences to be aligned, and let $P[1..r]$ be a given pattern constraining the alignments. The algorithm modifies the dynamic-programming solution of the ordinary multiple sequence alignment (MSA). It adds a new dimension of size $r+1$ such that each position k, $0 \le k \le r$, on this dimension can be considered as a layer that corresponds to the ordi-

nary MSA matrix for sequences S_1, S_2, ..., S_n. The case when k=0 is the ordinary MSA. Alignments at layer k satisfy the constraint for the pattern sequence $P[1..k]$. An optimum alignment is obtained at layer $k=r$ in $O(2^n s_1 s_2 \ldots s_n)$ time.

There are several improved sequential and parallel algorithms for the CMSA problem (e.g. He, Arslan, & Ling, 2006).

Regular Expression Motif

The constraint for the sequence alignment problem can also be a regular expression. Arslan (2007) introduced the regular expression constrained sequence alignment (RECSA) as the following problem: given strings S_1, S_2, and a regular expression R, find the maximum alignment score between S_1 and S_2 over all alignments that satisfy a given regular expression constraint described by R. An alignment satisfies the constraint if it includes a segment in which a substring s_1 of S_1 is aligned to a substring s_2 of S_2, and both s_1 and s_2 match R. We can verify that the alignment in Fig. 1 satisfies the constraint if the given regular expression is $R=G(\epsilon+\Sigma)\Sigma CT$ where ϵ denotes the null-string, and Σ denotes the alphabet over which sequences are defined. This alignment includes a region where the substring GCCA of S_1 is aligned to substring GTTCA of S_2, and both substrings match R.

Considering common structures is especially important in aligning protein sequences. Family of protein sequences include conserved regions (motifs) which are associated with particular functions. Typically, motifs span 10 to 30 symbols where each symbol represents one of the 20 amino acids. These motifs are described in PROSITE format (http://www.expasy.org/txt/prosuser.txt). For example, in PROSITE, the famous P-loop motif (PS00017) is represented as *[GA]-X(4)-G-K-[ST]*, which means that the first position of the motif can be occupied by the symbol A or G, the second, third, fourth, and fifth positions can be occupied by any symbol (*X(4)* denotes four consecutive wildcards each of which matches all symbols), and the sixth and seventh positions have to be G and L, respectively, followed by either S or T.

Arslan (2007) presents an algorithm for the RECSA problem. He constructs a nondeterministic finite automaton M from a given motif, and modifies the underlying dynamic programming solution for the sequence alignment problem so that instead of optimal scores, optimal weights for states in M are computed. Arslan's algorithm runs in $O(t^4nm)$ time where t is the number of states in a nondeterministic finite automaton recognizing the given motif, and n and m are the lengths of the sequences aligned. Chung, Lu, & Tang (2007) present a faster, $O(t^3nm)$ time, algorithm for the problem. They also give a $O(t^2 \log t\, n)$-time algorithm when $t = O(\log n)$ where n is the common length of the sequences aligned.

FUTURE TRENDS

Using regular expression motifs in guiding alignments gives rise to several computationally challenging problems attracting further research. The regular expression constrained sequence alignment problem for multiple sequences is computationally expensive. There have been several attempts to solve this problem using various methods. Chung et al. (2007) uses the *progressive sequence alignment* technique (Durbin et al., 1998) for this case. Progressive alignment algorithms for the constrained sequence alignment problem when the constraint is inclusion of a subsequence were proposed earlier (e.g. Tang et al., 2002). Progressive alignment is an approximation for the original problem. It first picks a pair of sequences, and aligns them. Then, a third sequence is chosen and aligned to the first alignment. This process continues until all sequences are aligned.

Another interesting variation of the regular expression constrained sequence alignment problem is the case when motifs are not readily given but they are any in PROSITE database, or when sequences share multiple motifs that may appear in any order. There are several algorithms addressing these cases (e.g. Arslan, 2007).

Comparing complex molecules such as RNA is a very involved process in which we need to take into account the physical structures of the molecules. There are many proposed models for RNA sequence, and structure comparisons. A comprehensive summary of these techniques is given in Chapter 4 of Wang et al. (2005).

Many interesting RNA molecules conserve a secondary structure of base-pairing interactions more than they conserve their sequences. These base-pairing interactions can be described by a *context free grammar*. *Stochastic context free grammars* (SCFG)s have been used in the description of RNA secondary structures

(Durbin et al., 1998). An SCFG is a CFG where productions have probabilities. There are many SCFG-based methods (Durbin et al., 1998). For example, Dowell and Eddy (2006) use pair stochastic context-free grammars, *pairSCFGs*, for pairwise structural alignment of RNA sequences with the constraint that the structures are confidently aligned at positions (pins) which are known a priori. Dowell and Eddy (2006) assume a general SCFG that works well for all RNA sequences.

Arslan (2007) considers RNA sequence alignment guided by motifs described by CFGs, and presents a method that assumes a known database of motifs each of which is represented by a context free grammar. This method proposes a dynamic programming solution that finds the maximum score obtained by an alignment that is composed of pairwise *motif-matching regions*, and ordinary pairwise sequence alignments of non-motif-matching regions. A motif-matching region is a pair *(x,y)* of regions *x*, and *y*, respectively, in the given two sequences aligned such that both *x* and *y* are generated by the same CFG *G* in the given database. Each such pair *(x,y)* has a score assigned to that *G* (if there are multiple such CFG's, then the maximum score over all such G's is taken). The main advantage of this approach over the existing CFG (or SCFG)-based algorithms is that when we guide alignments by given motifs, base-pairing positions are not fixed in any of the sequences; there may be many candidate pairwise matches for common (sub)structures (determined by a given set of motifs) in both sequences, and there may be in various possible orders; one that yields the optimum score is chosen.

We project that with the increasing amount of raw biological data, the databases that store common structures will also grow rapidly. Sequence alignments, and discovering common sequence structures from sequences will go hand-in-hand affecting one another. We have witnessed this trend already. The notion of motif was first explicitly introduced by Russell Doolittle (1981). Discovery of sequence motifs has continued at a steady rate, and the motifs, in the form of amino acid patterns, were incorporated by Amos Bairoch in the PROSITE database (http://www.expasy.org/prosite). For many years, PROSITE (Falquet et al., 2002) has been a collection of thousands of sequence motifs. Now, this knowledge is used in regular expression constrained sequence alignments. A similar trend can be seen for the discovery of structural motifs identifiable in sequences. A large number of RNA motifs that bind proteins are already known (Burd and Dreyfus, 1994). New motifs continue to be discovered (Klein et al., 2001). Motifs that can be identified in sequences will be very helpful in guiding sequence alignment.

There are several topics that we did not cover in this chapter. Protein sequence motifs can also be described by a matrix in which each entry gives the probability of a given symbol appearing in a given position. We note that this definition can also be used to describe a constraint in sequence alignment.

Motif discovery is a very important area outside the scope of this chapter. We direct the reader to Wang et al. (2005) (Section 2.4) for a list of techniques on motif discovery, and Lu, R., Jia, C., Zhang S., Chen L., & Zhang, H. (2007) for a recent paper.

We think that the following are important future computational challenges in aligning sequences under the guidance of motifs:

- developing new representations for complex common structures (motifs),
- discovering new motifs from sequences, and building databases out of them,
- using known motif databases to guide alignments when multiple motifs are shared (in various possible orders) in sequences aligned.

CONCLUSION

Knowledge discovery from sequences yields better sequence alignment tools that use this knowledge in guiding the sequence alignment process. Better sequence alignments in turn yield more knowledge that can be fed into this loop. Sequence alignment has become an increasingly more knowledge-guided optimization problem.

REFERENCES

Arslan, A. N. (2007). Sequence alignment guided by common motifs described by context free grammars. The *4th Annual Biotechnology and Bioinformatics Symposium (BIOT-07)* Colorado Springs, CO, Oct. 19-20.

Arslan, A. N. (2007). Regular expression constrained sequence alignment. *Journal of Discrete Algorithms*, 5(4),

647-661. (available online: http://dx.doi.org/10.1016/j.jda.2007.01.003) (preliminary version appeared in the *16th Annual Combinatorial Pattern Matching (CPM) Symposium*, Jeju Island, Korea, *LNCS 3537*, 322-333, June 19-22, 2005)

Arslan, A. N., & Eğecioğlu, Ö. (2005). Algorithms for the constrained common subsequence problem. *International Journal of Foundations of Computer Science*, 16(6), 1099–1109.

Arslan, A. N., Eğecioğlu, Ö., & Pevzner, P. A. (2001). A new approach to sequence comparison: normalized local alignment. *Bioinformatics*, 17, 327–337.

Burd, C. G., & Dreyfus, G. (1994). Conserved structures and diversity of functions of RNA binding motifs. *Science*, 265,615-621.

Chin, F. Y. L., Ho, N. L., Lam, T. W., Wong, P. W. H., & Chan, M. Y. (2003). Efficient constrained multiple sequence alignment with performance guarantee. *Proc. IEEE Computational Systems Bioinformatics (CSB 2003)*, 337-346.

Chung, Y.-S., Lu, C. L., & Tang, C. Y. (2007). Efficient algorithms for regular expression constrained sequence alignment. *Information Processing Letters*, 103(6), 240-246.

Chung, Y.-S., Lee, W.-H., Tang, C. Y., & Lu, C. L. (in press). RE-MuSiC: a tool for multiple sequence alignment with regular expression constraints. Nucleic Acids Research (available online: doi:10.1093/nar/gkm275)

Comet, J.-P., & Henry, J. (2002). Pairwise sequence alignment using a PROSITE pattern-derived similarity score. *Computers and Chemistry*, 26, 421-436.

Dowell, R. D., & Eddy, S. R. (2006). Efficient pairwise RNA structure prediction and alignment using sequence alignment constraints. *BMC Bioinformatics*,7(400):1-18.

Doolittle, R. F. (1981). Similar amino acid sequences: chance or common ancestry. *Science*, 214, 149-159.

Durbin, R., Eddy, S., Krogh, A., & Michison, G. (1998). *Biological sequence analysis*. *Cambridge* Cambridge, UK: University Press.

Falquet, L., Pagni, M., Bucher, P., Hulo, N., Sigrist, C. J., Hofmann, K., & Bairoch, A. (2002). The PROSITE database, its status in 2002. *Nucleic Acids Research*, 30, 235–238.

He, D., Arslan, A. N., & Ling, A. C. H. (2006). A fast algorithm for the constrained multiple sequence alignment problem. *Acta Cybernetica*, 17, 701–717.

Huang, X. (1994). A context dependent method for comparing sequences. *Combinatorial Pattern Matching (CPM) Symposium*, *LNCS 807*, 54–63.

Klein, D. J., Schmeing, T. M., Moore, P. B. & Steitz, T. A. (2001). The kink-turn: a new RNA secondary structure motif. *The EMBO Journal*, 20(15):4214-4221.

Lu, R., Jia, C., Zhang S., Chen L., & Zhang, H. (2007). An exact data mining method for finding center strings and their instances. *IEEE Transactions on Knowledge and Data Engineering*, 19(4):509-522.

Tang, C. Y., Lu, C. L., Chang, M. D.-T., Tsai, Y.-T., Sun, Y.-J., Chao, K.-M., Chang, J.-M., Chiou, Y.-H., Wu, C.-M., Chang, H.-T., & Chou, W.-I. (2002). Constrained multiple sequence alignment tool development and its applications to RNase family alignment. In Proc. of *CSB 2002*, 127-137.

Wang, J. T. L. , Zaki, M. J., Toivonen, H. T. T., & Shasha, (2005). D. (Eds). *Data mining in bioinformatics*, London, UK: Springer.

Zhang,Z., Berman,P., & Miller,W. (1998). Alignments without low scoring regions. *Journal of Computational Biology*, 5, 197–200.

KEY TERMS

Biological Sequences: DNA, and RNA sequences are strings of 4 nucleotides. Protein sequences are strings of 20 amino acids.

Constrained Sequence Alignment: Sequence alignment constrained to contain a given pattern. The pattern can be described in various ways. For example, it can be a string that will be contained as a subsequence, or a regular expression.

Context-Free Grammar: A 4-tuple (V,Σ,R,S) where V is a finite set of variables, Σ is a finite set of terminals disjoint from V, S in V is the start variable, and R is the set of rules of the for $A \rightarrow w$ where A is in

V, and *w* is in $(V \cup T)^*$, i.e. *w* is a string of variables and terminals.

Motif: A biological pattern in sequences that indicates evolutionary relatedness, or common function.

PROSITE Pattern: A regular expression-like pattern with two main differences from an ordinary regular expression: it does not contain the Kleene Closure *, and *X(i,j)* is used to denote wildcards whose lengths vary between *i, j*. Every PROSITE pattern can be converted to an equivalent regular expression.

Regular Expression: A null-string, a single symbol, or an empty set, or recursively defined as follows: if *R* is a regular expression so is *R**, and if *R* and *S* are regular expressions then so are *(R U S)*, and *RS*.

Scoring Matrix: A substitution matrix that assigns to every pair of amino acids a similarity score. There are several such matrices, e.g. PAM, BLOSUM.

Sequence Alignment: A scheme for writing sequences one under another to maximize the total column scores.

Subsequence of String *S*. A string that is obtained from *S* after deleting symbols from *S*.

Stochastic Grammar: A grammar whose rules have associated probabilities.

Hierarchical Document Clustering

Benjamin C. M. Fung
Concordia University, Canada

Ke Wang
Simon Fraser University, Canada

Martin Ester
Simon Fraser University, Canada

INTRODUCTION

Document clustering is an automatic grouping of text documents into clusters so that documents within a cluster have high similarity in comparison to one another, but are dissimilar to documents in other clusters. Unlike document classification (Wang, Zhou, & He, 2001), no labeled documents are provided in clustering; hence, clustering is also known as unsupervised learning. Hierarchical document clustering organizes clusters into a tree or a hierarchy that facilitates browsing. The parent-child relationship among the nodes in the tree can be viewed as a topic-subtopic relationship in a subject hierarchy such as the Yahoo! directory.

This chapter discusses several special challenges in hierarchical document clustering: high dimensionality, high volume of data, ease of browsing, and meaningful cluster labels. State-of-the-art document clustering algorithms are reviewed: the partitioning method (Steinbach, Karypis, & Kumar, 2000), agglomerative and divisive hierarchical clustering (Kaufman & Rousseeuw, 1990), and frequent itemset-based hierarchical clustering (Fung, Wang, & Ester, 2003). The last one, which was developed by the authors, is further elaborated since it has been specially designed to address the hierarchical document clustering problem.

BACKGROUND

Document clustering is widely applicable in areas such as search engines, web mining, information retrieval, and topological analysis. Most document clustering methods perform several preprocessing steps including stop words removal and stemming on the document set. Each document is represented by a vector of frequencies of remaining terms within the document. Some document clustering algorithms employ an extra preprocessing step that divides the actual term frequency by the overall frequency of the term in the entire document set. The idea is that if a term is too common across different documents, it has little discriminating power (Rijsbergen, 1979). Although many clustering algorithms have been proposed in the literature, most of them do not satisfy the special requirements for clustering documents:

- **High dimensionality.** The number of relevant terms in a document set is typically in the order of thousands, if not tens of thousands. Each of these terms constitutes a dimension in a document vector. Natural clusters usually do not exist in the full dimensional space, but in the subspace formed by a set of correlated dimensions. Locating clusters in subspaces can be challenging.
- **Scalability.** Real world data sets may contain hundreds of thousands of documents. Many clustering algorithms work fine on small data sets, but fail to handle large data sets efficiently.
- **Accuracy.** A good clustering solution should have high intra-cluster similarity and low inter-cluster similarity, i.e., documents within the same cluster should be similar but are dissimilar to documents in other clusters. An external evaluation method, the F-measure (Rijsbergen, 1979), is commonly used for examining the accuracy of a clustering algorithm.
- **Easy to browse with meaningful cluster description.** The resulting topic hierarchy should provide a sensible structure, together with meaningful cluster descriptions, to support interactive browsing.
- **Prior domain knowledge.** Many clustering algorithms require the user to specify some input

parameters, e.g., the number of clusters. However, the user often does not have such prior domain knowledge. Clustering accuracy may degrade drastically if an algorithm is too sensitive to these input parameters.

MAIN FOCUS

Hierarchical Clustering Methods

One popular approach in document clustering is agglomerative hierarchical clustering (Kaufman & Rousseeuw, 1990). Algorithms in this family build the hierarchy bottom-up by iteratively computing the similarity between all pairs of clusters and then merging the most similar pair. Different variations may employ different similarity measuring schemes (Zhao & Karypis, 2001; Karypis, 2003). Steinbach (2000) shows that Unweighted Pair Group Method with Arithmatic Mean (UPGMA) (Kaufman & Rousseeuw, 1990) is the most accurate one in its category. The hierarchy can also be built top-down which is known as the divisive approach. It starts with all the data objects in the same cluster and iteratively splits a cluster into smaller clusters until a certain termination condition is fulfilled.

Methods in this category usually suffer from their inability to perform adjustment once a merge or split has been performed. This inflexibility often lowers the clustering accuracy. Furthermore, due to the complexity of computing the similarity between every pair of clusters, UPGMA is not scalable for handling large data sets in document clustering as experimentally demonstrated in (Fung, Wang, & Ester, 2003).

Partitioning Clustering Methods

K-means and its variants (Larsen & Aone, 1999; Kaufman & Rousseeuw, 1990; Cutting, Karger, Pedersen, & Tukey, 1992) represent the category of partitioning clustering algorithms that create a flat, non-hierarchical clustering consisting of k clusters. The k-means algorithm iteratively refines a randomly chosen set of k initial centroids, minimizing the average distance (i.e., maximizing the similarity) of documents to their closest (most similar) centroid. The bisecting k-means algorithm first selects a cluster to split, and then employs basic k-means to create two sub-clusters, repeating these two steps until the desired number k of clusters is reached. Steinbach (2000) shows that the bisecting k-means algorithm outperforms basic k-means as well as agglomerative hierarchical clustering in terms of accuracy and efficiency (Zhao & Karypis, 2002).

Both the basic and the bisecting k-means algorithms are relatively efficient and scalable, and their complexity is linear to the number of documents. As they are easy to implement, they are widely used in different clustering applications. A major disadvantage of k-means, however, is that an incorrect estimation of the input parameter, the number of clusters, may lead to poor clustering accuracy. Also, the k-means algorithm is not suitable for discovering clusters of largely varying sizes, a common scenario in document clustering. Furthermore, it is sensitive to noise that may have a significant influence on the cluster centroid, which in turn lowers the clustering accuracy. The k-medoids algorithm (Kaufman & Rousseeuw, 1990; Krishnapuram, Joshi, & Yi, 1999) was proposed to address the noise problem, but this algorithm is computationally much more expensive and does not scale well to large document sets.

Frequent Itemset-Based Methods

Wang et al. (1999) introduced a new criterion for clustering transactions using frequent itemsets. The intuition of this criterion is that many frequent items should be shared within a cluster while different clusters should have more or less different frequent items. By treating a document as a transaction and a term as an item, this method can be applied to document clustering; however, the method does not create a hierarchy of clusters.

The Hierarchical Frequent Term-based Clustering (HFTC) method proposed by (Beil, Ester, & Xu, 2002) attempts to address the special requirements in document clustering using the notion of frequent itemsets. HFTC greedily selects the next frequent itemset, which represents the next cluster, minimizing the overlap of clusters in terms of shared documents. The clustering result depends on the order of selected itemsets, which in turn depends on the greedy heuristic used. Although HFTC is comparable to bisecting k-means in terms of clustering accuracy, experiments show that HFTC is not scalable (Fung, Wang, Ester, 2003).

A Scalable Algorithm for Hierarchical Document Clustering: FIHC

A scalable document clustering algorithm, Frequent Itemset-based Hierarchical Clustering (FIHC) (Fung, Wang, & Ester, 2003), is discussed in greater detail because this method satisfies all of the requirements of document clustering mentioned above. We use "item" and "term" as synonyms below. In classical hierarchical and partitioning methods, the pairwise similarity between documents plays a central role in constructing a cluster; hence, those methods are "document-centered". FIHC is "cluster-centered" in that it measures the cohesiveness of a cluster directly using frequent itemsets: documents in the same cluster are expected to share more common itemsets than those in different clusters.

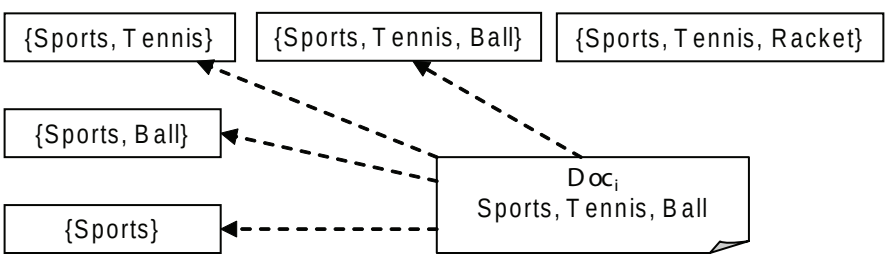
A frequent itemset is a set of terms that occur together in some minimum fraction of documents. To illustrate the usefulness of this notion for the task of clustering, let us consider two frequent items, "windows" and "apple". Documents that contain the word "windows" may relate to renovation. Documents that contain the word "apple" may relate to fruits. However, if both words occur together in many documents, then another topic that talks about operating systems should be identified. By precisely discovering these hidden topics as the first step and then clustering documents based on them, the quality of the clustering solution can be improved. This approach is very different from HFTC where the clustering solution greatly depends on the order of selected itemsets. Instead, FIHC assigns documents to the best cluster from among all available clusters (frequent itemsets). The intuition of the clustering criterion is that there are some frequent itemsets for each cluster in the document set, but different clusters share few frequent itemsets. FIHC uses frequent itemsets to construct clusters and to organize clusters into a topic hierarchy.

The following definitions are introduced in (Fung, Wang, Ester, 2003): A *global frequent itemset* is a set of items that appear together in more than a minimum fraction of the whole document set. A *global frequent item* refers to an item that belongs to some global frequent itemset. A global frequent itemset containing *k* items is called a *global frequent k-itemset*. A global frequent item is *cluster frequent* in a cluster C_i if the item is contained in some minimum fraction of documents in C_i. FIHC uses only the global frequent items in document vectors; thus, the dimensionality is significantly reduced.

The FIHC algorithm can be summarized in three phases: First, construct initial clusters. Second, build a cluster (topic) tree. Finally, prune the cluster tree in case there are too many clusters.

1. **Constructing clusters:** For each global frequent itemset, an initial cluster is constructed to include all the documents containing this itemset. Initial clusters are overlapping because one document may contain multiple global frequent itemsets. FIHC utilities this global frequent itemset as the cluster label to identify the cluster. For each document, the "best" initial cluster is identified and the document is assigned only to the best matching initial cluster. The goodness of a cluster C_i for a document doc_j is measured by some score function using cluster frequent items of initial clusters. After this step, each document belongs to exactly one cluster. The set of clusters can be viewed as a set of topics in the document set.
   Example: Figure 1 depicts a set of initial clusters. Each of them is labeled with a global frequent itemset. A document Doc_1 containing global

Figure 1. Initial clusters



frequent items "Sports", "Tennis", and "Ball" is assigned to clusters {Sports}, {Sports, Ball}, {Sports, Tennis} and {Sports, Tennis, Ball}. Suppose {Sports, Tennis, Ball} is the "best" cluster for Doc_1 measured by some score function. Doc_1 is then removed from {Sports}, {Sports, Ball}, and {Sports, Tennis}.

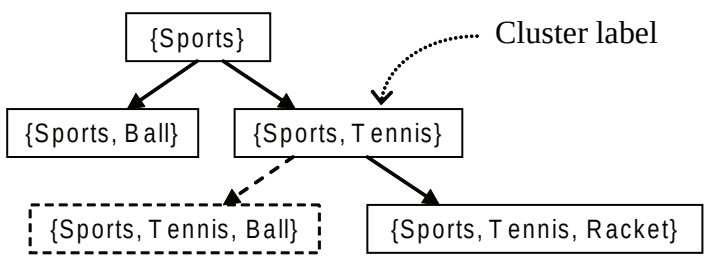
2. **Building cluster tree:** In the cluster tree, each cluster (except the root node) has exactly one parent. The topic of a parent cluster is more general than the topic of a child cluster and they are "similar" to a certain degree (see Figure 2 for an example). Each cluster uses a global frequent *k*-itemset as its cluster label. A cluster with a *k*-itemset cluster label appears at level *k* in the tree. The cluster tree is built bottom up by choosing the "best" parent at level *k-1* for each cluster at level *k*. The parent's cluster label must be a subset of the child's cluster label. By treating all documents in the child cluster as a single document, the criterion for selecting the best parent is similar to the one for choosing the best cluster for a document.
   Example: Cluster {Sports, Tennis, Ball} has a global frequent 3-itemset label. Its potential parents are {Sports, Ball} and {Sports, Tennis}. Suppose {Sports, Tennis} has a higher score. It becomes the parent cluster of {Sports, Tennis, Ball}.
3. **Pruning cluster tree:** The cluster tree can be broad and deep, which becomes not suitable for browsing. The goal of tree pruning is to efficiently remove the overly specific clusters based on the notion of inter-cluster similarity. The idea is that if two sibling clusters are very similar, they should be merged into one cluster. If a child cluster is very similar to its parent (high inter-cluster similarity), then replace the child cluster with its parent cluster. The parent cluster will then also include all documents of the child cluster.
   Example: Suppose the cluster {Sports, Tennis, Ball} is very similar to its parent {Sports, Tennis} in Figure 2. {Sports, Tennis, Ball} is pruned and its documents, e.g., Doc_1, are moved up into cluster {Sports, Tennis}.

Evaluation of FIHC

The FIHC algorithm was experimentally evaluated and compared to state-of-the-art document clustering methods. See (Fung, Wang, & Ester, 2003) for more details. FIHC uses only the global frequent items in document vectors, drastically reducing the dimensionality of the document set. Experiments show that clustering with reduced dimensionality is significantly more efficient and scalable. FIHC can cluster 100K documents within several minutes while HFTC and UPGMA cannot even produce a clustering solution. FIHC is not only scalable, but also accurate. The clustering accuracy of FIHC, evaluated based on F-measure (Steinbach, Karypis, & Kumar, 2000), consistently outperforms other methods. FIHC allows the user to specify an optional parameter, the desired number of clusters in the solution. However, close-to-optimal accuracy can still be achieved even if the user does not specify this parameter.

The cluster tree provides a logical organization of clusters which facilitates browsing documents. Each cluster is attached with a cluster label that summarizes the documents in the cluster. Different from other clustering methods, no separate post-processing is required for generating these meaningful cluster descriptions.

Figure 2. Sample cluster tree



Related Links

The followings are some clustering tools on the Internet:

Tools: FIHC implements Frequent Itemset-based Hierarchical Clustering.
Website: http://www.cs.sfu.ca/~ddm/
Tools: CLUTO implements Basic/Bisecting K-means and Agglomerative methods.
Website: http://glaros.dtc.umn.edu/gkhome/views/cluto/
Tools: Vivísimo® is a clustering search engine.
Website: http://vivisimo.com/

FUTURE TRENDS

Incrementally Updating the Cluster Tree

One potential research direction is to incrementally update the cluster tree (Guha, Mishra, Motwani, & O'Callaghan, 2000; Hennig & Wurst, 2006; Murua, Stuetzle, Tantrum & Sieberts, 2008). In many cases, the number of documents is growing continuously in a document set, and it is infeasible to rebuild the cluster tree upon every arrival of a new document. Using FIHC, one can simply assign new documents to the most similar existing cluster. The clustering accuracy, however, may degrade in the course of the time since the original global frequent itemsets may no longer reflect the current state of the overall document set. Incremental clustering is closely related to some of the recent research on data mining in stream data (Ordonez, 2003).

Incorporate with Natural Language Processing Techniques

Many existing document clustering algorithms consider a document as a bag of words. Although the semantic relationships among the words may be crucial for clustering, they are not utilized. Li and Chung (2005) considered a sequence of words is frequent if it occurs in more than certain percentage of the documents in a document set. Wang and Hodges (2006) proposed to use the sense disambiguation method to identify the sense of words to construct the feature vector for document representation. Future work could incorporate the natural language processing technique and utilize the Universal Networking Language (Uchida, Zhu, & Della Senta, 2000), a semantic representation for sentences, to generate feature vectors and computing scores.

CONCLUSION

Most traditional clustering methods do not address the special requirements for hierarchical document clustering, including high dimensionality, high volume, and ease of browsing. In this chapter, we review several document clustering methods in the context of these requirements, and a document clustering method, FIHC, is discussed a detail. Due to massive volumes of unstructured data generated in the globally networked environment, the importance of document clustering will continue to grow.

REFERENCES

Beil, F., Ester, M., & Xu, X. (2002). Frequent term-based text clustering. *International Conference on Knowledge Discovery and Data Mining*, KDD'02, Edmonton, Alberta, Canada, 436-442.

Cutting, D. R., Karger, D. R., Pedersen, J. O., & Tukey, J. W. (1992). Scatter/gather: A cluster-based approach to browsing large document collections. *International Conference on Research and Development in Information Retrieval*, SIGIR'92, Copenhagen, Denmark, 318–329.

Fung, B. C. M., Wang, K., & Ester, M. (2003, May). Hierarchical document clustering using frequent itemsets. *SIAM International Conference on Data Mining*, SDM'03, San Francisco, CA, United States, 59-70.

Guha, S., Mishra, N., Motwani, R., & O'Callaghan, L. (2000). Clustering data streams. *Symposium on Foundations of Computing Science*, 359-366.

Hennig, S & Wurst, M. (2006). Incremental Clustering of Newsgroup Articles. International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE'06, 332-341.

Kaufman, L., & Rousseeuw, P. J. (1990, March). *Finding Groups in Data: An Introduction to Cluster Analysis*. New York: John Wiley & Sons, Inc.

Karypis, G. (2003). Cluto 2.1.1: A Software Package for Clustering High Dimensional Datasets. The Internet < http://glaros.dtc.umn.edu/gkhome/views/cluto/>

Krishnapuram, R., Joshi, A., & Yi, L. (1999, August). A fuzzy relative of the k-medoids algorithm with application to document and snippet clustering. *IEEE International Conference - Fuzzy Systems*, FUZZIEEE 99, Korea.

Larsen, B., & Aone, C. (1999). Fast and effective text mining using linear-time document clustering. *International Conference on Knowledge Discovery and Data Mining*, KDD'99, San Diego, California, United States, 16–22.

Li, Y. & Chung, S. M. (2005). Text document clustering based on frequent word sequences. *International Conference on Information and Knowledge Management*, CIKM '05, Bremen, Germany, 293-294.

Murua, A., Stuetzle, W., Tantrum, J., and Sieberts, S. (2008). Model Based Document Classification and Clustering. International Journal of Tomography and Statistics, 8(W08).

Ordonez, C. (2003). Clustering binary data streams with K-means. *Workshop on Research issues in data mining and knowledge discovery*, SIGMOD'03, San Diego, California, United States, 12-19.

van Rijsbergen, C. J. (1979). *Information Retrieval*. London: Butterworth Ltd., second edition.

Steinbach, M., Karypis, G., & Kumar, V. (2000). A comparison of document clustering techniques. *Workshop on Text Mining*, SIGKDD'00.

Uchida, H., Zhu, M., & Della Senta, T. (2000) A gift for a millennium. The United Nations University, 2000.

Wang, K., Xu, C., & Liu, B. (1999). Clustering transactions using large items. *International Conference on Information and Knowledge Management*, CIKM'99, Kansas City, Missouri, United States, 483–490.

Wang, K., Zhou, S., & He, Y. (2001, Apr.). Hierarchical classification of real life documents. *SIAM International Conference on Data Mining*, SDM'01, Chicago, United States.

Wang, Y. & Hodges, J. (2006). Document Clustering with Semantic Analysis. Hawaii International Conference on System Sciences (HICSS'06) Track 3, 54c.

Zhao, Y., & Karypis, G. (2001). Criterion functions for document clustering: Experiments and analysis. Technical report, Department of Computer Science, University of Minnesota.

Zhao, Y., & Karypis, G. (2002, Nov.). Evaluation of hierarchical clustering algorithms for document datasets. *International Conference on Information and Knowledge Management*, McLean, Virginia, United States, 515-524.

KEY TERMS

Cluster Frequent Item: A global frequent item is cluster frequent in a cluster C_i if the item is contained in some minimum fraction of documents in C_i.

Document Clustering: The automatic organization of documents into clusters or group so that documents within a cluster have high similarity in comparison to one another, but are very dissimilar to documents in other clusters.

Document Vector: Each document is represented by a vector of frequencies of remaining items after preprocessing within the document.

Global Frequent Itemset: A set of words that occur together in some minimum fraction of the whole document set.

Inter-cluster Similarity: The overall similarity among documents from two different clusters.

Intra-cluster Similarity: The overall similarity among documents within a cluster.

Medoid: The most centrally located object in a cluster.

Stop Words Removal: A preprocessing step for text mining. Stop words, like "the" and "this" which rarely help the mining process, are removed from input data.

Stemming: For text mining purposes, morphological variants of words that have the same or similar semantic interpretations can be considered as equivalent. For example, the words "computation" and "compute" can be stemmed into "comput".

Histograms for OLAP and Data-Stream Queries

Francesco Buccafurri
DIMET, Università di Reggio Calabria, Italy

Gianluca Caminiti
DIMET, Università di Reggio Calabria, Italy

Gianluca Lax
DIMET, Università di Reggio Calabria, Italy

INTRODUCTION

Histograms are an important tool for data reduction both in the field of data-stream querying and in OLAP, since they allow us to represent large amount of data in a very compact structure, on which both efficient mining techniques and OLAP queries can be executed. Significant time- and memory-cost advantages may derive from data reduction, but the trade-off with the accuracy has to be managed in order to obtain considerable improvements of the overall capabilities of mining and OLAP tools.

In this chapter we focus on histograms, that are shown in the recent literature to be one of the possible concrete answers to the above requirements.

BACKGROUND

Data synopses are widely exploited in many applications. Every time it is necessary to produce fast query answers and a certain estimation error can be accepted, it is possible to inquire summary data rather than the original ones and to perform suitable interpolations. This happens for example in OLAP, where a typical query is a *range query*, or in the case of continuous query over data streams.

A possible solution to this problem is using sampling methods (Gemulla, Lehner, & Haas, 2007; Gryz, Guo, Liu & Zuzarte, 2004): only a small number of suitably selected records of R, well *representing* R, are stored. The query is then evaluated by exploiting these samples instead of the full relation R. Sampling techniques are very easy to implement.

Regression techniques try to model data as a function in such a way that only a small set of coefficients representing such a function is stored, rather than the original data. The simplest regression technique is the linear one, modeling a data distribution as a linear function. Despite its simplicity, not allowing to capture complex relationships among data, this technique often produces acceptable results. There are also non-linear regressions, significantly more complex than the linear one from the computational point of view, yet applicable to a much larger set of cases.

Besides these techniques, another possible solution relies on the usage of histograms.

MAIN THRUST OF THE CHAPTER

Histograms are a lossy compression technique widely used in various application contexts, like query optimization, statistical and temporal databases and OLAP applications. In OLAP, compression allows us to obtain fast approximate answers by evaluating queries on reduced data in place of the original ones. Histograms are well-suited to this purpose, especially in case of range queries (Muthukrishnan & Strauss, 2003).

A histogram is a compact representation of a relation R. It is obtained by partitioning an attribute X of the relation R into k sub-ranges, called buckets, and by maintaining for each of them a few information, typically corresponding to the bucket boundaries, the number of tuples with value of X belonging to the sub-range associated to the bucket (often called *sum of the bucket*), and the number of distinct values of X of such a sub-range occurring in some tuple of R (i.e., the number of non-null frequencies of the sub-range).

Figure 1 reports an example of 3-bucket histogram, built on a domain of size 12 with 3 null elements. For each bucket (represented as an oval), we have reported the boundaries (on the left and the right side, respectively) and the value of the sum of the elements belonging to the bucket (inside the oval). Observe that, the null values (i.e. the values at 6, 7 and 9) do not occur in any bucket.

A range query, defined on an interval I of X, evaluates the number of occurrences in R with value of X in I. Thus, buckets embed a set of pre-computed disjoint range queries capable of covering the whole active domain of X in R (by "active" here we mean attribute values actually appearing in R). As a consequence, the histogram does not give, in general, the possibility of evaluating exactly a range query not corresponding to one of the pre-computed embedded queries. In other words, while the contribution to the answer coming from the sub-ranges coinciding with entire buckets can be returned exactly, the contribution coming from the sub-ranges which partially overlap buckets can be only estimated, since the actual data distribution inside the buckets is not available. For example, concerning the histogram shown in Figure 1, a range query from 4 to 8 is estimated by summing (1) the partial contribution of bucket 1 computed by CVA (see Section "Estimation inside a bucket"), that is 104.8, and (2) the *sum* of bucket 2, that is 122. As a consequence, the range query estimation is 226.8 whereas the exact result is 224.

Constructing the best histogram means defining the boundaries of buckets in such a way that the estimation of the non pre-computed range queries becomes more effective (e.g., by avoiding that large frequency differences arise inside a bucket). This approach corresponds to finding, among all possible sets of pre-computed range queries, the set which guarantees the best estimation of the other (non pre-computed) queries, once a technique for estimating such queries is defined.

Besides this problem, which we call the *partition problem*, there is another relevant issue to investigate: How to improve the estimation inside the buckets? We discuss about both the above issues in the following two sections.

The Partition Problem

This issue has been widely analyzed in the past and a number of techniques have been proposed. Among these, we first consider the Max-Diff histogram and the V-Optimal histogram. Even though they are not the most recent techniques, we deeply cite them since they are still considered points of reference.

We start by describing the Max-Diff histogram.

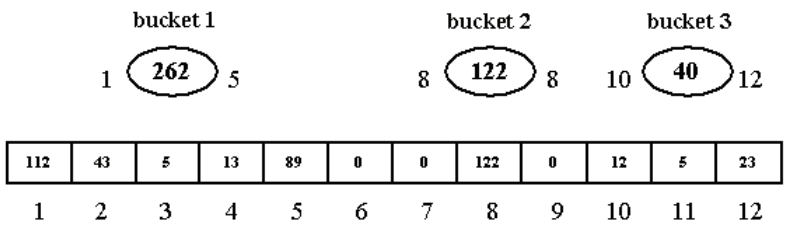
Let $V=\{v_1, \ldots, v_n\}$ be the set of values of the attribute X actually appearing in the relation R and $f(v_i)$ be the number of tuples of R having value v_i in X. A Max-Diff histogram with h buckets is obtained by putting a boundary between two adjacent attribute values v_i and v_{i+1} of V if the difference between $f(v_{i+1}) \cdot s_{i+1}$ and $f(v_i) \cdot s_i$ is one of the $h-1$ largest such differences (where s_i denotes the spread of v_i, that is the distance from v_i to the next non-null value).

A V-Optimal histogram, which is the other classical histogram we describe, produces more precise results than the Max-Diff histogram. It is obtained by selecting the boundaries for each bucket i so that the query approximation error is minimal. In particular, the boundaries of each bucket i, say lb_i and ub_i (with $1 \le i \le h$, where h is the total number of buckets), are fixed in such a way that:

$$\sum_{i=1}^{h} SSE_i$$

is minimum, where the standard squared error of the i-th bucket

Figure 1. An example of a 3-bucket histogram built on a domain of size 12.



$$SSE_i = \sum_{j=lb_i}^{ub_i} (f(j) - avg_i)^2$$

where *f(j)* is the frequency of the *j*-th value of the domain and avg_i is the average of the frequencies occurring in the *i*-th bucket. A V-Optimal histogram uses a dynamic programming technique in order to find the "optimal" partitioning w.r.t. the given error metrics.

Even though the V-Optimal histogram is more accurate than Max-Diff, its high space- and time-complexities make it rarely used in practice.

In order to overcome such a drawback, an approximate version of the V-Optimal histogram has been proposed. The basic idea is quite simple: First, data are partitioned into *l* disjoint chunks and then the V-Optimal algorithm is used in order to compute a histogram within each chuck. The consequent problem is how to allocate buckets to the chunks such that exactly *B* buckets are used. This is solved by implementing a dynamic programming scheme. It is shown that an approximate V-Optimal histogram with *B*+*l* buckets has the same accuracy as the non-approximate V-Optimal with *B* buckets. Moreover, the time required for executing the approximate algorithm is reduced by a multiplicative factor equal to1/*l*.

Besides the guarantee of accuracy, the histogram should efficiently support hierarchical range queries in order not to limit too much the capability of drilling-down and rolling-up over data.

This requirement was introduced by Koudas, Muthukrishnan & Srivastava (2000), who have shown the insufficient accuracy of classical histograms in evaluating hierarchical range queries. Therein, the Hierarchical Optimal histogram *H-Optimal* is proposed, that has the minimum total expected error for estimating prefix range queries, where the only queries allowed are one-sided ranges. The error metrics used is the SSE, described above for V-Optimal. Moreover, a polynomial-time algorithm for its construction is provided.

Guha, Koudas & Srivastava (2002) have proposed efficient algorithms for the problem of approximating the distribution of measure attributes organized into hierarchies. Such algorithms are based on dynamic programming and on a notion of sparse intervals.

Guha, Shim & Woo (2004) have proposed the Relative Error Histogram *RE-HIST*. It is the optimal histogram that minimizes the relative error measures for estimating prefix range queries. The relative error for a range query is computed as $(x_i\text{-}e_i)/\max\{|x_i|,c\}$ where x_i and e_i are the actual value and the estimation of the range query, respectively, and c is a *sanity constant* (typically fixed to 1), used to reduce excessive domination of relative error by small data values.

Buccafurri & Lax (2004a) have presented a histogram based on a hierarchical decomposition of the data distribution kept in a full binary tree. Such a tree, containing a set of pre-computed hierarchical queries, is encoded by using bit saving for obtaining a smaller structure and, thus, for efficiently supporting hierarchical range queries.

All the histograms presented above are widely used as synopses for persistent data, that is data rarely modified, like databases or data warehouses. However, more sophisticated histograms can be exploited also for *data streams*.

Data streams is an emergent issue that in the last years has captured the interest of many scientific communities. The crucial problem, arising in several application contexts like network monitoring, sensor networks, financial applications, security, telecommunication data management, Web applications, and so on, is dealing with continuous data flows (i.e. data streams) having the following characteristics: (1) they are time-dependent, (2) their size is very large, so that they cannot be totally stored due to the actual memory limitation, and (3) data arrival is very fast and unpredictable, so that each data management operation should be very efficient.

Since a data stream consists of a large amount of data, it is usually managed on the basis of a *sliding window*, including only the most recent data. Thus, any technique capable of compressing sliding windows by maintaining a good approximate representation of data distribution is relevant in this field. Typical queries performed on sliding windows are *similarity queries* and other analysis, like *change mining queries* (Dong, Han, Lakshmanan, Pei, Wang & Yu, 2003) useful for trend analysis and, in general, for understanding the dynamics of data. Also in this field, histograms may become an important analysis tool. The challenge is finding new histograms that (1) are fast to construct and to maintain, that is the required updating operations (performed at each data arrival) are very efficient, (2) maintain a good accuracy in approximating data distribution and (3) support continuous querying on data.

An example of the above emerging approaches is reported in (Buccafurri & Lax, 2004b), where a tree-like histogram with cyclic updating is proposed. By using

such a compact structure, many mining techniques, that would take very high computational costs when used on real data streams, can be effectively implemented. Guha, Koudas & Shim (2006) have presented a one-pass linear-time approximation algorithm for the histogram construction problem for a wide variety of error measures. Due to their fast construction, the histograms here proposed can be profitably extended to the case of data streams.

Besides bucket-based histograms, there are other kinds of histograms whose construction is not driven by the search of a suitable partition of the attribute domain and, further, their structure is more complex than simply a set of buckets. This class of histograms is called *non-bucket based* histograms. *Wavelets* are an example of such kind of histograms.

Wavelets are mathematical transformations storing data in a compact and hierarchical fashion, used in many application contexts, like image and signal processing (Khalifa, 2003; Kacha, Grenez, De Doncker & Benmahammed, 2003). There are several types of transformations, each belonging to a family of wavelets. The result of each transformation is a set of values, called *wavelet coefficients*. The advantage of this technique is that, typically, the value of a (possibly large) number of wavelet coefficients results to be below a fixed threshold, so that such coefficients can be approximated by 0. Clearly, the overall approximation of the technique as well as the compression ratio depends on the value of such a threshold. In the last years wavelets have been exploited in data mining and knowledge discovery in databases, thanks to time-and space-efficiency and data hierarchical decomposition characterizing them (Garofalakis & Kumar, 2004; Guha, Chulyun & Shim, 2004; Garofalakis & Gibbons, 2002; Karras & Mamoulis, 2005, Garofalakis & Gibbons, 2004).

In the next section we deal with the second problem introduced earlier, concerning the estimation of range queries partially involving buckets.

Estimation Inside a Bucket

While finding the optimal bucket partition has been widely investigated in the past years, the problem of estimating queries partially involving a bucket has received a little attention.

Histograms are well-suited to range query evaluation, since buckets basically correspond to a set of precomputed range queries. A range query that involves entirely one or more buckets can be computed exactly, while if it overlaps partially a bucket, then the result can be only estimated.

The simplest estimation technique is the Continuous Value Assumption (CVA): Given a bucket of size s and sum c, a range query overlapping the bucket in i points is estimated as $(i/s)\cdot c$. This corresponds to estimating the partial contribution of the bucket to the range query result by linear interpolation.

Another possibility is to use the Uniform Spread Assumption (USA). It assumes that values are distributed at equal distance from each other and the overall frequency sum is equally distributed among them. In this case, it is necessary to know the number of non-null frequencies belonging to the bucket. Denoting by t such a value, the range query is estimated by:

$$\frac{(s-1)\cdot(i-1)\cdot(t-1)}{(s-1)}\cdot\frac{c}{t}.$$

An interesting problem is understanding whether, by exploiting information typically contained in histogram buckets, and possibly adding some concise summary information, the frequency estimation inside buckets, and then, the histogram accuracy, can be improved. To this aim, starting from a theoretical analysis about limits of CVA and USA, Buccafurri, Lax, Pontieri, Rosaci & Saccà (2008) have proposed to use an additional storage space of 32 bits (called 4LT) in each bucket, in order to store the approximate representation of the data distribution inside the bucket. 4LT is used to save approximate cumulative frequencies at 7 equi-distant intervals internal to the bucket.

Clearly, approaches similar to that followed in (Buccafurri et al., 2008) have to deal with the trade-off between the extra storage space required for each bucket and the number of total buckets the allowed total storage space consents.

FUTURE TRENDS

The main limit of the first decade of research in querying-oriented histograms was certainly that they did not provide guarantees about the approximation error. The first significant work where the goal of error guarantee is reached, under some conditions, is by Datar, Gionis, Indyk & Motwani (2002). In particular, the

proposed histogram gives the error guarantee only in case of biased range queries (which are significant in the context of data streams), i.e. involving the last N data of the sliding window, for a given query size N. However, in the general case, the form of the queries cannot be limited to this kind of query (like in KDD applications). The current and future challenge is thus to find new histograms with good error-guarantee features and with no limitation about the kind of queries which can be approximated.

CONCLUSION

In many application contexts like OLAP and data streams the usage of histograms is becoming more and more frequent. Indeed they allow us to perform OLAP queries on summarized data in place of the original ones, or to arrange effective approaches requiring multiple scans on data streams, that, in such a way, may be performed over a compact version of the data stream. In these cases, the usage of histograms allows us to analyze a large amount of data very fast, guaranteeing at the same time good accuracy requirements.

REFERENCES

Buccafurri, F., Lax, G., Pontieri, L., Rosaci, D., & Saccà, D. (2008). Enhancing Histograms by Tree-Like Bucket Indices. The *VLDB Journal, The International Journal on Very Large Data Bases.*

Buccafurri, F., & Lax, G. (2004a). Fast range query estimation by N-level tree histograms. *Data & Knowledge Engineering Journal*, Volume 58, Issue 3, p 436-465, Elsevier Science.

Buccafurri, F., & Lax, G. (2004b). Reducing Data Stream Sliding Windows by Cyclic Tree-Like Histograms. In *Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases*, 75-86.

Datar, M., Gionis, A., Indyk, P., & Motwani, R. (2002). Maintaining Stream Statistics over Sliding Windows. *SIAM Journal on Computing*, Volume 31, Issue 6, 1794-1813.

Dong, G., Han, J., Lakshmanan, L. V. S., Pei, J., Wang, H., & Yu, P. S. (2003). Online mining of changes from data streams: Research problems and preliminary results. In *Proceedings of the ACM SIGMOD Workshop on Management and Processing of Data Streams*.

Garofalakis, M., & Gibbons, P. B. (2002). Wavelet Synopses with Error Guarantees. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 476-487.

Garofalakis, M., & Gibbons, P. BP. (2004). Probabilistic wavelet synopses. *ACM Transaction on Database Systems*, Volume 29, Issue 1, 43-90.

Garofalakis, M., & Kumar, A. (2004). Deterministic Wavelet Thresholding for Maximum Error Metrics. In *Proceedings of the Twenty-third ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, 166-176.

Gemulla, R., Lehner, W., & Haas, P. J. (2007). Maintaining bernoulli samples over evolving multisets. In *Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, 93-102.

Gryz, J., Guo, J., Liu, L., & Zuzarte, C. (2004). Query sampling in DB2 Universal Database. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, 839-843.

Guha, S., Koudas, N., & Srivastava, D. (2002). Fast algorithms for hierarchical range histogram construction. In *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, 180-187.

Guha, S., Shim, K., & Woo, J. (2004). REHIST: Relative error histogram construction algorithms. In *Proceedings of the Thirtieth International Conference on Very Large Data Bases*, 300-311.

Guha, S., K., Chulyun & Shim, K. (2004). XWAVE: Approximate Extended Wavelets for Streaming Data. In *Proceedings of the International Conference on Very Large Data Bases*, 288-299.

Guha, S., Koudas, N., & Shim, K. (2006). Approximation and streaming algorithms for histogram construction problems. *ACM Transaction on Database Systems*, Volume 31, Issue 1, 396-438.

Kacha, A., Grenez, F., De Doncker, P., & Benmahammed, K. (2003). A wavelet-based approach for frequency estimation of interference signals in printed

circuit boards. In *Proceedings of the 1st international symposium on Information and communication technologies*, 21-26.

Karras, P. & Mamoulis, N. (2005). One-pass wavelet synopses for maximum-error metrics. In *Proceedings of the International Conference on Very Large Data Bases*, 421-432.

Khalifa, O. (2003). Image data compression in wavelet transform domain using modified LBG algorithm. In *Proceedings of the 1st international symposium on Information and communication technologies*, 88-93.

Koudas, N., Muthukrishnan, S., & Srivastava, D. (2000). Optimal histograms for hierarchical range queries. In *Proceedings of the nineteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, 196–204.

Muthukrishnan, S., & Strauss, M. (2003). Rangesum histograms. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, 233-242.

KEY TERMS

Bucket: An element obtained by partitioning the domain of an attribute X of a relation into non-overlapping intervals. Each bucket consists of a tuple <*inf, sup, val*> where *val* is an aggregate information (for an instance sum, average, count, and so on) about tuples with value of X belonging to the interval (*inf, sup*).

Bucket-Based Histogram: A type of histogram whose construction is driven by searching for a suitable partition of the attribute domain into buckets.

Continuous Query: In the context of data stream it is a query issued once and running continuously over the data.

Data Stream: Data that is structured and processed in a continuous flow, like digital audio and video or data coming from digital sensors.

Histogram: A set of buckets implementing a partition of the overall domain of a relation attribute.

Range Query: A query returning an aggregate information (such as sum, average) about data belonging to a given interval of the domain.

Similarity Query: A kind of query aiming to discovering correlations and patterns in dynamic data streams and widely used to perform trend analysis.

Sliding Window: A sequence of the most recent values, arranged in arrival time order, that are collected from a data stream.

Wavelets: Mathematical transformations implementing hierarchical decomposition of functions leading to the representation of functions through sets of wavelet coefficients.

Homeland Security Data Mining and Link Analysis

Bhavani Thuraisingham
The MITRE Corporation, USA

INTRODUCTION

Data mining is the process of posing queries to large quantities of data and extracting information often previously unknown using mathematical, statistical, and machine-learning techniques. Data mining has many applications in a number of areas, including marketing and sales, medicine, law, manufacturing, and, more recently, homeland security. Using data mining, one can uncover hidden dependencies between terrorist groups as well as possibly predict terrorist events based on past experience. One particular data-mining technique that is being investigated a great deal for homeland security is link analysis, where links are drawn between various nodes, possibly detecting some hidden links.

This article provides an overview of the various developments in data-mining applications in homeland security. The organization of this article is as follows. First, we provide some background on data mining and the various threats. Then, we discuss the applications of data mining and link analysis for homeland security. Privacy considerations are discussed next as part of future trends. The article is then concluded.

BACKGROUND

We provide background information on both data mining and security threats.

Data Mining

Data mining is the process of posing various queries and extracting useful information, patterns, and trends often previously unknown from large quantities of data possibly stored in databases. Essentially, for many organizations, the goals of data mining include improving marketing capabilities, detecting abnormal patterns, and predicting the future, based on past experiences and current trends. There is clearly a need for this technology. There are large amounts of current and historical data being stored. Therefore, as databases become larger, it becomes increasingly difficult to support decision making. In addition, the data could be from multiple sources and multiple domains. There is a clear need to analyze the data to support planning and other functions of an enterprise.

Some of the data-mining techniques include those based on statistical reasoning techniques, inductive logic programming, machine learning, fuzzy sets, and neural networks, among others. The data-mining problems include classification (finding rules to partition data into groups), association (finding rules to make associations between data), and sequencing (finding rules to order data). Essentially, one arrives at some hypotheses, which is the information extracted from examples and patterns observed. These patterns are observed from posing a series of queries; each query may depend on the responses obtained to the previous queries posed.

Data mining is an integration of multiple technologies. These include data management, such as database management, data warehousing, statistics, machine learning, decision support, and others, such as visualization and parallel computing. There is a series of steps involved in data mining. These include getting the data organized for mining, determining the desired outcomes to mining, selecting tools for mining, carrying out the mining, pruning the results so that only the useful ones are considered further, taking actions from the mining, and evaluating the actions to determine benefits. There are various types of data mining. By this we do not mean the actual techniques used to mine the data, but what the outcomes will be. These outcomes also have been referred to as data-mining tasks. These include clustering, classification anomaly detection, and forming associations.

While several developments have been made, there also are many challenges. For example, due to the large volumes of data, how can the algorithms determine

which technique to select and what type of data mining to do? Furthermore, the data may be incomplete and/or inaccurate. At times, there may be redundant information, and at times, there may not be sufficient information. It is also desirable to have data-mining tools that can switch to multiple techniques and support multiple outcomes. Some of the current trends in data mining include mining Web data, mining distributed and heterogeneous databases, and privacy-preserving data mining, where one ensures that one can get useful results from mining and at the same time maintain the privacy of individuals (Berry & Linoff; Han & Kamber, 2000; Thuraisingham, 1998).

Security Threats

Security threats have been grouped into many categories (Thuraisingham, 2003). These include information-related threats, where information technologies are used to sabotage critical infrastructures, and non-information-related threats, such as bombing buildings. Threats also may be real-time threats and non-real-time threats. Real-time threats are threats where attacks have timing constraints associated with them, such as "building X will be attacked within three days." Non-real-time threats are those threats that do not have timing constraints associated with them. Note that non-real-time threats could become real-time threats over time.

Threats also include bioterrorism, where biological and possibly chemical weapons are used to attack, and cyberterrorism, where computers and networks are attacked. Bioterrorism could cost millions of lives, and cyberterrorism, such as attacks on banking systems, could cost millions of dollars. Some details on the threats and countermeasures are discussed in various texts (Bolz, 2001). The challenge is to come up with techniques to handle such threats. In this article, we discuss data-mining techniques for security applications.

MAIN THRUST

First, we will discuss data mining for homeland security. Then, we will focus on a specific data-mining technique called *link analysis* for homeland security. An aspect of homeland security is cyber security. Therefore, we also will discuss data mining for cyber security.

Applications of Data Mining for Homeland Security

Data-mining techniques are being examined extensively for homeland security applications. The idea is to gather information about various groups of people and study their activities and determine if they are potential terrorists. As we have stated earlier, data-mining outcomes include making associations, linking analyses, forming clusters, classification, and anomaly detection. The techniques that result in these outcomes are techniques based on neural networks, decisions trees, market-basket analysis techniques, inductive logic programming, rough sets, link analysis based on graph theory, and nearest-neighbor techniques. The methods used for data mining include top-down reasoning, where we start with a hypothesis and then determine whether the hypothesis is true, or bottom-up reasoning, where we start with examples and then come up with a hypothesis (Thuraisingham, 1998). In the following, we will examine how data-mining techniques may be applied for homeland security applications. Later, we will examine a particular data-mining technique called *link analysis* (Thuraisingham, 2003).

Data-mining techniques include techniques for making associations, clustering, anomaly detection, prediction, estimation, classification, and summarization. Essentially, these are the techniques used to obtain the various data-mining outcomes. We will examine a few of these techniques and show how they can be applied to homeland security. First, consider association rule mining techniques. These techniques produce results, such as John and James travel together or Jane and Mary travel to England six times a year and to France three times a year. Essentially, they form associations between people, events, and entities. Such associations also can be used to form connections between different terrorist groups. For example, members from Group A and Group B have no associations, but Groups A and B have associations with Group C. Does this mean that there is an indirect association between A and C?

Next, let us consider clustering techniques. Clusters essentially partition the population based on a characteristic such as spending patterns. For example, those living in the Manhattan region form a cluster, as they spend over $3,000 on rent. Those living in the Bronx from another cluster, as they spend around $2,000 on rent. Similarly, clusters can be formed based on terrorist activities. For example, those living in region X bomb

buildings, and those living in region Y bomb planes.

Finally, we will consider anomaly detection techniques. A good example here is learning to fly an airplane without wanting to learn to take off or land. The general pattern is that people want to get a complete training course in flying. However, there are now some individuals who want to learn to fly but do not care about take off or landing. This is an anomaly. Another example is John always goes to the grocery store on Saturdays. But on Saturday, October 26, 2002, he went to a firearms store and bought a rifle. This is an anomaly and may need some further analysis as to why he is going to a firearms store when he has never done so before. Some details on data mining for security applications have been reported recently (Chen, 2003).

Applications of Link Analysis

Link analysis is being examined extensively for applications in homeland security. For example, how do we connect the dots describing the various events and make links and connections between people and events? One challenge to using link analysis for counterterrorism is reasoning with partial information. For example, agency A may have a partial graph, agency B another partial graph, and agency C a third partial graph. The question is how do you find the associations between the graphs, when no agency has the complete picture? One would ague that we need a data miner that would reason under uncertainty and be able to figure out the links between the three graphs. This would be the ideal solution, and the research challenge is to develop such a data miner. The other approach is to have an organization above the three agencies that will have access to the three graphs and make the links.

The strength behind link analyses is that by visualizing the connections and associations, one can have a better understanding of the associations among the various groups. Associations such as A and B, B and C, D and A, C and E, E and D, F and B, and so forth can be very difficult to manage, if we assert them as rules. However, by using nodes and links of a graph, one can visualize the connections and perhaps draw new connections among different nodes. Now, in the real world, there would be thousands of nodes and links connecting people, groups, events, and entities from different countries and continents as well as from different states within a country. Therefore, we need link analysis techniques to determine the unusual connection, such as a connection between G and P, for example, which is not obvious with simple reasoning strategies or by human analysis.

Link analysis is one of the data-mining techniques that is still in its infancy. That is, while much has been written about techniques such as association rule mining, automatic clustering, classification, and anomaly detection, very little material has been published on link analysis. We need interdisciplinary researchers such as mathematicians, computational scientists, computer scientists, machine-learning researchers, and statisticians working together to develop better link analysis tools.

Applications of Data Mining for Cyber Security

Data mining also has applications in cyber security, which is an aspect of homeland security. The most prominent application is in intrusion detection. For example, our computers and networks are being intruded by unauthorized individuals. Data-mining techniques, such as those for classification and anomaly detection, are being used extensively to detect such unauthorized intrusions. For example, data about normal behavior is gathered, and when something occurs out of the ordinary, it is flagged as an unauthorized intrusion. Normal behavior could be that John's computer is never used between 2:00 A.M. and 5:00 A.M.. When John's computer is in use at 3:00 A.M., for example, then this is flagged as an unusual pattern.

Data mining is also being applied to other applications in cyber security, such a auditing. Here again, data on normal database access is gathered, and when something unusual happens, then this is flagged as a possible access violation. Digital forensics is another area where data mining is being applied. Here again, by mining the vast quantities of data, one could detect the violations that have occurred. Finally, data mining is being used for biometrics. Here, pattern recognition and other machine-learning techniques are being used to learn the features of a person and then to authenticate the person, based on the features.

FUTURE TRENDS

While data mining has many applications in homeland security, it also causes privacy concerns. This is

because we need to collect all kinds of information about people, which causes private information to be divulged. Privacy and data mining have been the subject of much debate during the past few years, although some early discussions also have been reported (Thuraisingham, 1996).

One promising direction is privacy-preserving data mining. The challenge here is to carry out data mining but at the same time ensure privacy. For example, one could use randomization as a technique and give out approximate values instead of the actual values. The challenge is to ensure that the approximate values are still useful. Many papers on privacy-preserving data mining have been published recently (Agrawal & Srikant, 2000).

CONCLUSION

This article has discussed data-mining applications in homeland security. Applications in national security and cyber security both are discussed. We first provided an overview of data mining and security threats and then discussed data-mining applications. We also emphasized a particular data-mining technique—link analysis. Finally, we discussed privacy-preserving data mining.

It is only during the past three years that data mining for security applications has received a lot of attention. Although a lot of progress has been made, there is also a lot of work that needs to be done. First, we need to have a better understanding of the various threats. We need to determine which data-mining techniques are applicable to which threats. Much research is also needed on link analysis. To develop effective solutions, data-mining specialists have to work with counter-terrorism experts. We also need to motivate the tool vendors to develop tools to handle terrorism.

NOTE

The views and conclusions expressed in this article are those of the author and do not reflect the policies of the MITRE Corporation or of the National Science Foundation.

REFERENCES

Agrawal, R, & Srikant, R. (2000). Privacy-preserving data mining. *Proceedings of the ACM SIGMOD Conference*, Dallas, Texas.

Berry, M., & Linoff, G. (1997). *Data mining techniques for marketing, sales, and customer support*. New York: John Wiley.

Bolz, F. (2001). *The counterterrorism handbook: Tactics, procedures, and techniques*. CRC Press.

Chen, H. (Ed.). (2003). *Proceedings of the 1st Conference on Security Informatics*, Tucson, Arizona.

Han, J., & Kamber, M. (2000). *Data mining, concepts and techniques.* CA: Morgan Kaufman.

Thuraisingham, B. (1996). Data warehousing, data mining and security. *Proceedings of the IFIP Database Security Conference*, Como, Italy.

Thuraisingham, B. (1998). *Data mining: Technologies, techniques, tools and trends*, FL: CRC Press.

Thuraisingham, B. (2003). *Web data mining technologies and their applications in business intelligence and counter-terrorism*. FL: CRC Press.

KEY TERMS

Cyber Security: Techniques used to protect the computer and networks from the threats.

Data Management: Techniques used to organize, structure, and manage the data, including database management and data administration.

Digital Forensics: Techniques to determine the root causes of security violations that have occurred in a computer or a network.

Homeland Security: Techniques used to protect a building or an organization from threats.

Intrusion Detection: Techniques used to protect the computer system or a network from a specific threat, which is unauthorized access.

Link Analysis: A data-mining technique that uses concepts and techniques from graph theory to make associations.

Privacy: Process of ensuring that information deemed personal to an individual is protected.

Privacy-Preserving Data Mining: Data-mining techniques that extract useful information but at the same time ensure the privacy of individuals.

Threats: Events that disrupt the normal operation of a building, organization, or network of computers.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 566-569, copyright 2005 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

Humanities Data Warehousing

Janet Delve
University of Portsmouth, UK

INTRODUCTION

Data Warehousing is now a well-established part of the business and scientific worlds. However, up until recently, data warehouses were restricted to modeling essentially numerical data – examples being sales figures in the business arena (in say Wal-Mart's data warehouse (Westerman, 2000)) and astronomical data (for example SKICAT) in scientific research, with textual data providing a descriptive rather than a central analytic role. The lack of ability of data warehouses to cope with mainly non-numeric data is particularly problematic for humanities[1] research utilizing material such as memoirs and trade directories. Recent innovations have opened up possibilities for 'non-numeric' data warehouses, making them widely accessible to humanities research for the first time. Due to its irregular and complex nature, humanities research data is often difficult to model, and manipulating time shifts in a relational database is problematic as is fitting such data into a normalized data model. History and linguistics are exemplars of areas where relational databases are cumbersome and which would benefit from the greater freedom afforded by data warehouse dimensional modeling.

BACKGROUND

Hudson (2001, p. 240) declared relational databases to be the predominant software used in recent, historical research involving computing. Historical databases have been created using different types of data from diverse countries and time periods. Some databases are modest and independent, others part of a larger conglomerate like the North Atlantic Population Project (NAPP) project that entails integrating international census data. One issue that is essential to good database creation is data modeling; which has been contentiously debated recently in historical circles.

When reviewing relational modeling in historical research, (Bradley, 1994) contrasted 'straightforward' business data with incomplete, irregular, complex- or semi-structured historical data. He noted that the relational model worked well for simply-structured business data, but could be tortuous to use for historical data. (Breure, 1995) pointed out the advantages of inputting data into a model that matches it closely, something that is very hard to achieve with the relational model. (Burt[2] and James, 1996) considered the relative freedom of using source-oriented data modeling (Denley, 1994) as compared to relational modeling with its restrictions due to normalization (which splits data into many separate tables), and highlighted the possibilities of data warehouses. Normalization is not the only hurdle historians encounter when using the relational model.

Date and time fields provide particular difficulties: historical dating systems encompass a number of different calendars, including the Western, Islamic, Revolutionary and Byzantine. Historical data may refer to 'the first Sunday after Michaelmas', requiring calculation before a date may be entered into a database. Unfortunately, some databases and spreadsheets cannot handle dates falling outside the late 20th century. Similarly, for researchers in historical geography, it might be necessary to calculate dates based on the local introduction of the Gregorian calendar, for example. These difficulties can be time-consuming and arduous for researchers. Awkward and irregular data with abstruse dating systems thus do not fit easily into a relational model that does not lend itself to hierarchical data. Many of these problems also occur in linguistics computing.

Linguistics is a data-rich field, with multifarious forms for words, multitudinous rules for coding sounds, words and phrases, and also numerous other parameters - geography, educational and social status. Databases are used for housing many types of linguistic data from a variety of research domains - phonetics, phonology, morphology, syntax, lexicography, computer-assisted learning (CAL), historical linguistics and dialectology. Data integrity and consistency are of utmost importance in this field. Relational DataBase Management Systems (RDBMSs) are able to provide this, together with powerful and flexible search facilities (Nerbonne, 1998, introduction).

Bliss and Ritter (2001 IRCS conference proceedings) discussed the constraints imposed on them when using 'the rigid coding structure of the database' developed to house pronoun systems from 109 languages. They observed that coding introduced interpretation of data and concluded that designing 'a typological database is not unlike trying to fit a square object into a round hole. Linguistic data is highly variable, database structures are highly rigid, and the two do not always "fit".' Brown (2001 IRCS conference proceedings) outlined the fact that different database structures may reflect a particular linguistic theory, and also mentioned the trade-off between quality and quantity in terms of coverage.

The choice of data model thus has a profound effect on the problems that can be tackled and the data that can be interrogated. For both historical and linguistic research, relational data modeling using normalization often appears to impose data structures which do not fit naturally with the data and which constrain subsequent analysis. Coping with complicated dating systems can also be very problematic. Surprisingly, similar difficulties have already arisen in the business community, and have been addressed by data warehousing.

MAIN THRUST

Data Warehousing in the Business Context

Data warehouses came into being as a response to the problems caused by large, centralized databases which users found unwieldy to query. Instead, they extracted portions of the databases which they could then control, resulting in the 'spider-web' problem where each department produces queries from its own, uncoordinated extract database (Inmon 2002, pp. 6-14). The need was thus recognized for a single, integrated source of clean data to serve the *analytical needs* of a company.

A data warehouse can provide answers to a completely different range of queries than those aimed at a traditional database. Using an estate agency as a typical business, the type of question their local databases should be able to answer might be 'How many three-bedroomed properties are there in the Botley area up to the value of £150,000?' The type of over-arching question a business analyst (and CEOs) would be interested in might be of the general form 'Which type of property sells for prices above the average selling price for properties in the main cities of Great Britain and how does this correlate to demographic data?' (Begg and Connolly, 2004, p. 1154). To trawl through each local estate agency database and corresponding local county council database, then amalgamate the results into a report would consume vast quantities of time and effort. The data warehouse was created to answer this type of need.

Basic Components of a Data Warehouse

Inmon (2002, p. 31), the 'father of data warehousing', defined a data warehouse as being *subject-oriented, integrated, non-volatile and time-variant.* Emphasis is placed on choosing the right *subjects* to model as opposed to being constrained to model around *applications*. Data warehouses do not replace databases as such - they co-exist alongside them in a symbiotic fashion. Databases are needed both to serve the clerical community who answer day-to-day queries such as 'what is A.R. Smith's current overdraft?' and also to 'feed' a data warehouse. To do this, snapshots of data are extracted from a database on a regular basis (daily, hourly and in the case of some mobile phone companies almost real-time). The data is then transformed (cleansed to ensure consistency) and loaded into a data warehouse. In addition, a data warehouse can cope with diverse data sources, including external data in a variety of formats and summarized data from a database. The myriad types of data of different provenance create an exceedingly rich and varied integrated data source opening up possibilities not available in databases. Thus all the data in a data warehouse is *integrated*. Crucially, data in a warehouse is not updated - it is only added to, thus making it *non-volatile*, which has a profound effect on data modeling, as the main function of normalization is to obviate update anomalies. Finally, a data warehouse has a time horizon (that is contains data over a period) of five to ten years, whereas a database typically holds data that is current for two to three months.

Data Modeling in a Data Warehouse: Dimensional Modeling

There is a fundamental split in the data warehouse community as to whether to construct a data warehouse from scratch, or to build them via data marts. A data mart is essentially a cut-down data warehouse that is

restricted to one department or one business process. Inmon (2002, p. 142) recommended building the data warehouse first, then extracting the data from it to fill up several data marts. The data warehouse modeling expert Kimball (2002) advised the incremental building of several data marts that are then carefully integrated into a data warehouse. Whichever way is chosen, the data is normally modeled via dimensional modeling. Dimensional models need to be linked to the company's corporate ERD (Entity Relationship Diagram) as the data is actually taken from this (and other) source(s). Dimensional models are somewhat different from ERDs, the typical star model having a central fact table surrounded by dimension tables. Kimball (2002, pp. 16-18) defined a fact table as 'the primary table in a dimensional model where the numerical performance measurements of the business are stored…Since measurement data is overwhelmingly the largest part of any data mart, we avoid duplicating it in multiple places around the enterprise.' Thus the fact table contains dynamic numerical data such as sales quantities and sales and profit figures. It also contains key data in order to link to the dimension tables. Dimension tables contain the textual descriptors of the business process being modeled and their depth and breadth define the analytical usefulness of the data warehouse. As they contain descriptive data, it is assumed they will not change at the same rapid rate as the numerical data in the fact table that will certainly change every time the data warehouse is refreshed. Dimension tables typically have 50-100 attributes (sometimes several hundreds) and these *are not usually normalized*. The data is often *hierarchical* in the tables and can be an accurate reflection of how data actually appears in its raw state (Kimball 2002, pp. 19-21). There is not the need to normalize as data is not updated in the data warehouse, although there are variations on the star model such as the snowflake and starflake models which allow varying degrees of normalization in some or all of their dimension tables. Coding is disparaged due to the long-term view that definitions may be lost and that the dimension tables should contain the fullest, most comprehensible descriptions possible (Kimball 2002, p. 21). The restriction of data in the fact table to numerical data has been a hindrance to academic computing. However, Kimball has recently developed 'factless' fact tables (Kimball 2002, p. 49) which do not contain measurements, thus opening the door to a much broader spectrum of possible data warehouses.

Applying the Data Warehouse Architecture to Historical and Linguistic Research

One of the major advantages of data warehousing is the enormous flexibility in modeling data. Normalization is no longer an automatic straightjacket and hierarchies can be represented in dimension tables. The expansive time dimension (Kimball 2002, p. 39) is a welcome by-product of this modeling freedom, allowing country-specific calendars, synchronization across multiple time zones and the inclusion of multifarious time periods. It is possible to add external data from diverse sources and summarized data from the source database(s). The data warehouse is built for analysis which immediately makes it attractive to humanities researchers. It is designed to continuously receive huge volumes (terabytes) of data, but is sensitive enough to cope with the idiosyncrasies of geographic location dimensions within GISs (Kimball, 2002, p. 227). Additionally a data warehouse has advanced indexing facilities that make it desirable for those controlling vast quantities of data. With a data warehouse it is theoretically possible to publish the 'right data' that has been collected from a variety of sources and edited for quality and consistency. In a data warehouse all data is collated so a variety of different subsets can be analyzed whenever required. It is comparatively easy to extend a data warehouse and add material from a new source. The data cleansing techniques developed for data warehousing are of interest to researchers, as is the tracking facility afforded by the meta data manager (Begg and Connolly 2004, pp. 1169-70).

In terms of using data warehouses 'off the shelf', some humanities research might fit into the 'numerical fact' topology, but some might not. The 'factless fact table' has been used to create several American university data warehouses, but expertise in this area would not be as widespread as that with normal fact tables. The whole area of data cleansing may perhaps be daunting for humanities researchers (as it is to those in industry). Ensuring vast quantities of data is clean and consistent may be an unattainable goal for humanities researchers without recourse to expensive data cleansing software (and may of course distort source data (Delve and Healey, 2005, p.110)). The data warehouse technology is far from easy and is based on having existing databases to extract from, hence double the work. It is unlikely that researchers would be taking

regular snapshots of their data, as occurs in industry, but they could equate to data sets taken at different periods of time to data warehouse snapshots (e.g. 1841 census, 1861 census). Whilst many data warehouses use familiar WYSIWYGs and can be queried with SQL-type commands, there is undeniably a huge amount to learn in data warehousing. Nevertheless, there are many areas in both linguistics and historical research where data warehouses may prove attractive.

FUTURE TRENDS

Data Warehouses and Linguistics Research

The problems related by Bliss and Ritter (2001 IRCS conference proceedings) concerning rigid relational data structures and pre-coding problems would be alleviated by data warehousing. Brown (2001 IRCS conference proceedings) outlined the dilemma arising from the alignment of database structures with particular linguistic theories, and also the conflict of quality and quantity of data. With a data warehouse there is room for both vast quantities of data and a plethora of detail. No structure is forced onto the data so several theoretical approaches can be investigated using the same data warehouse. Dalli (2001 IRCS conference proceedings) observed that many linguistic databases are standalone with no hope of interoperability. His proffered solution to create an interoperable database of Maltese linguistic data involved an RDBMS and XML. Using data warehouses to store linguistic data should ensure interoperability. There is growing interest in corpora databases, with the dedicated conference at Portsmouth, November 2003. Teich, Hansen and Fankhauser drew attention to the multi-layered nature of corpora and speculated as to how 'multi-layer corpora can be maintained, queried and analyzed in an integrated fashion.' A data warehouse would be able to cope with this complexity.

Nerbonne (1998) alluded to the 'importance of coordinating the overwhelming amount of work being done and yet to be done.' Kretzschmar (2001 IRCS conference proceedings) delineated 'the challenge of preservation and display for massive amounts of survey data.' There appears to be many linguistics databases containing data from a range of locations / countries. For example, ALAP, the American Linguistic Atlas Project; ANAE, the Atlas of North American English (part of the TELSUR Project); TDS, the Typological Database System containing European data; AMPER, the Multimedia Atlas of the Romance Languages. Possible research ideas for the future may include a broadening of horizons - instead of the emphasis on individual database projects, there may develop an 'integrated warehouse' approach with the emphasis on larger scale, collaborative projects. These could compare different languages or contain many different types of linguistic data for a particular language, allowing for new orders of magnitude analysis.

Data Warehouses and Historical Research

There are inklings of historical research involving data warehousing in Britain and Canada. A data warehouse of current census data is underway at the University of Guelph, Canada and the Canadian Century Research Infrastructure aims to house census data from the last 100 years in data marts constructed using IBM software at several sites based in universities across the country. At the University of Portsmouth, UK, a historical data warehouse of American mining data was created by Professor Richard Healey using Oracle Warehouse Builder (Delve, Healey and Fletcher, 2004), and (Delve and Healey, 2005, p109-110) expound the research advantages conferred by the data warehousing architecture used. In particular, dimensional modeling opened up the way to powerful OLAP querying which in turn led to new areas of investigation as well as contributing to traditional research (Healey, 2007). One of the resulting unexpected avenues of research pursued was that of spatial data warehousing, and (Healey and Delve, 2007) delineate significant research results achieved by the creation of a Census data warehouse of the US 1880 census which integrated GIS and data warehousing in a web environment, utilizing a factless fact table at the heart of its dimensional model. They conclude that 'the combination of original person level, historical census data with a properly designed spatially-enabled data warehouse framework confirms the important warehouse principle that the basic data should be stored in the most disaggregated manner possible. Appropriate hierarchical levels specified within the dimensions then give the maximum scope to the OLAP mechanism for providing an immensely rich and varied analytical capability that goes far beyond

anything previously seen in terms of census analysis' (Healey and Delve, 2007, p.33) and also "the most novel contribution of GIS to data warehousing identified to date is that it facilitates the utilisation of linked spatial data sets to provide ad hoc spatially referenced query dimensions that can feed into the OLAP pipeline" (Healey and Delve, 2007, p.34), and all carried out in real-time. These projects give some idea of the scale of project a data warehouse can cope with, that is, really large country / state -wide problems. Following these examples, it would be possible to create a data warehouse to analyze all British censuses from 1841 to 1901 (approximately 10^8 bytes of data). Data from a variety of sources over time such as hearth tax, poor rates, trade directories, census, street directories, wills and inventories, GIS maps for a city such as Winchester could go into a city data warehouse. Similarly, a Voting data warehouse could contain voting data – poll book data and rate book data up to 1870 for the whole country. A Port data warehouse could contain all data from portbooks for all British ports together with yearly trade figures. Similarly a Street directories data warehouse would contain data from this rich source for whole country for the last 100 years. Lastly, a Taxation data warehouse could afford an overview of taxation of different types, areas or periods. The fact that some educational institutions have Oracle site licenses opens to way for humanities researchers with Oracle databases to use Oracle Warehouse Builder as part of the suite of programs available to them. These are practical project suggestions which would be impossible to construct using relational databases, but which, if achieved, could grant new insights into our history. Comparisons could be made between counties and cities and much broader analysis would be possible than has previously been the case.

CONCLUSION

The advances made in business data warehousing are directly applicable to many areas of historical and linguistics research. Data warehouse dimensional modeling allows historians and linguists to model vast amounts of data on a countrywide basis (or larger), incorporating data from existing databases, spreadsheets and GISs. Summary data could also be included, and this would all lead to a data warehouse containing more data than is currently possible, plus the fact that the data would be richer than in current databases due to the fact that normalization is no longer obligatory. Whole data sources can be captured, and more post-hoc analysis is a direct result. Dimension tables particularly lend themselves to hierarchical modeling, so data does not need splitting into many tables thus forcing joins while querying. The time dimension particularly lends itself to historical research where significant difficulties have been encountered in the past. These suggestions for historical and linguistics research clearly resonate in other areas of humanities research, such as historical geography, and any literary or cultural studies involving textual analysis (for example biographies, literary criticism and dictionary compilation).

REFERENCES

Begg, C. & Connolly, T. (2004). Database Systems. Harlow: Addison-Wesley.

Bliss and Ritter, IRCS (Institute for Research into Cognitive Science) Conference Proceedings (2001). The Internet <http://www.ldc.upenn.edu/annotation/database/proceedings.html>

Bradley, J. (1994). Relational Database Design. History and Computing, 6(2), 71-84.

Breure, L. (1995). Interactive data Entry. History and Computing, 7(1), 30-49.

Brown, IRCS Conference Proceedings (2001). The Internet <http://www.ldc.upenn.edu/annotation/database/proceedings.html>

Burt[ii], J. & James, T. B. (1996). Source-Oriented Data Processing: The triumph of the micro over the macro? History and Computing, 8(3), 160-169.

Dalli, IRCS Conference Proceedings (2001). The Internet <http://www.ldc.upenn.edu/annotation/database/proceedings.html>

Delve, J., Healey, R., & Fletcher, A. (2004). 'Teaching Data Warehousing as a Standalone Unit using Oracle Warehouse Builder', Proceedings of the British Computer Society TLAD Conference, July 2004, Edinburgh, UK.

Delve, J., & Healey, R. (2005). 'Is there a role for data warehousing technology in historical research?', Humanities, Computers and Cultural Heritage: Pro-

ceedings of the XVIth International Conference of the Association for History and Computing, Royal Netherlands Academy of Arts and Sciences, Amsterdam. 106-111.

Delve, J., & Allen, M. (2006). 'Large-scale integrated historical projects – does Data Warehousing offer any scope for their creation and analysis? History and Computing, Edinburgh Press. 301-313.

Denley, P. (1994). Models, Sources and Users: Historical Database Design in the 1990s. History and Computing, 6(1), 33-43.

Healey, R. (2007). The Pennsylvanian Anthracite Coal Industry 1860-1902. Scranton: Scranton University Press.

Healey, R., & Delve, J. (2007). 'Integrating GIS and data warehousing in a Web environment: A case study of the US 1880 Census', International Journal of Geographical Information Science, 21(6), 603-624.

Hudson, P. (2000). History by numbers. An introduction to quantitative approaches. London: Arnold.

Inmon, W. H. (2002). Building the Data Warehouse. New York: Wiley.

Kimball, R. & Ross, M. (2002). The Data Warehouse Toolkit. New York: Wiley.

Kretzschmar, IRCS Conference Proceedings (2001). The Internet <http://www.ldc.upenn.edu/annotation/database/proceedings.html>

NAPP website http://www.nappdata.org/napp/

Nerbonne, J. (ed) (1998). Linguistic Databases. Stanford Ca.: CSLI Publications.

SKICAT, The Internet <http://www-aig.jpl.nasa.gov/public/mls/home/fayyad/SKICAT-PR1-94.html>

Teich, Hansen & Fankhauser, IRCS Conference Proceedings (2001). The Internet <http://www.ldc.upenn.edu/annotation/database/proceedings.html>

Wal-Mart, the Internet http://www.tgc.com/dsstar/99/0824/100966.html

Westerman, P. (2000). Data Warehousing: Using the Wal-Mart Model. San Fransisco: Morgan Kaufmann

KEY TERMS

Data Modeling: The process of producing a model of a collection of data which encapsulates its semantics and hopefully its structure.

Dimensional Model: The dimensional model is the data model used in data warehouses and data marts, the most common being the star schema, comprising a fact table surrounded by dimension tables.

Dimension Table: Dimension tables contain the textual descriptors of the business process being modeled and their depth and breadth define the analytical usefulness of the data warehouse.

Fact Table: 'the primary table in a dimensional model where the numerical performance measurements of the business are stored' (Kimball, 2002).

Factless Tact Table: A fact table which contains no measured facts (Kimball 2002).

Normalization: The process developed by E.C. Codd whereby each attribute in each table of a relational database depend entirely on the key(s) of that table. As a result, relational databases comprise many tables, each containing data relating to one entity.

Spatial Data Warehouse: A data warehouse that can also incorporate Geographic Information System (GIS)-type queries.

ENDNOTES

1 Learning or literature concerned with human culture (Compact OED)

2 Now Delve

Hybrid Genetic Algorithms in Data Mining Applications

Sancho Salcedo-Sanz
Universidad de Alcalá, Spain

Gustavo Camps-Valls
Universitat de València, Spain

Carlos Bousoño-Calzón
Universidad Carlos III de Madrid, Spain

INTRODUCTION

G**enetic algorithms** (**GAs**) are a class of problem solving techniques which have been successfully applied to a wide variety of hard problems (Goldberg, 1989). In spite of conventional GAs are interesting approaches to several problems, in which they are able to obtain very good solutions, there exist cases in which the application of a conventional GA has shown poor results. Poor performance of GAs completely depends on the problem. In general, problems severely *constrained* or problems with difficult *objective functions* are hard to be optimized using GAs. Regarding the difficulty of a problem for a GA there is a well established theory. Traditionally, this has been studied for binary encoded problems using the so called Walsh Transform (Liepins & Vose, 1991), and its associated *spectrum* (Hordijk & Stadler, 1998), which provides an idea of the distribution of the important schemas (building blocks) in the search space.

Several methods to enhance the performance of GAs in difficult applications have been developed. Firstly, the encoding of a problem determines the search space where the GA must work. Therefore, given a problem, the selection of the best *encoding* is an important pre-processing step. *Operators* which reduce the search space are then interesting in some applications. Secondly, variable length or transformed encodings are schemes, which can be successfully applied to some difficult problems. The hybridization of a GA with *local search* algorithms can also improve the performance of the GA in concrete applications. There are two types of hybridization:

- If the GA is hybridized with a local search heuristic in order to tackle the problem constraints, it is usually known as a *hybrid genetic algorithm*.
- If the GA is hybridized with a local search heuristic in order to improve its performance, then it is known as a *memetic algorithm*.

In this chapter we revise several hybrid methods involving GAs that have been applied to data mining problems. First, we provide a brief background with several important definitions on genetic algorithms, hybrid algorithms and operators for improving its performance. In the Main Trust section, we present a survey of several hybrid algorithms, which use GAs as search heuristic, and their main applications in data mining. Finally, we finish the chapter giving some conclusions and future trends.

BACKGROUND

Genetic Algorithms

Genetic algorithms are robust problems' solving techniques based on natural evolution processes. They are population-based algorithms, which encode a set of possible solutions to the problem, and evolve it through the application of the so called *genetic operators* (Goldberg, 1989). The standard genetic operators in a GA are:

- *Selection*, where the individuals of a new population are selected from the old one. In the standard implementation of the Selection operator, each individual has a probability of surviving for the

next generation proportional to its associated fitness (objective function) value. This procedure of selection is usually called roulette wheel selection mechanism.

- *Crossover*, where new individuals are searched starting from couples of individuals in the population. Once the couples are randomly selected, the individuals have the possibility of swapping parts of themselves with its couple, the probability of this happens is usually called *crossover probability*, *P_c*.
- *Mutation*, where new individuals are searched by randomly changing bits of current individuals with a low probability *P_m* (*probability of mutation*).

Operators for Enhancing Genetic Algorithms' Performance

Enhancing genetic algorithms with different new operators or local search algorithms to improve their performance in difficult problems is not a new topic. From the beginning of the use of these algorithms, researchers noted that there were applications too difficult to be successfully solved by the conventional GA. Thus, the use of new operators and the hybridization with other heuristics were introduced as mechanisms to improve GAs performance in many difficult problems.

The use of new operators is almost always forced by the problem encoding, usually in problems with constraints. The idea is to use the local knowledge about the problem by means of introducing these operators. Usually they are special *Crossover* and *Mutation* operators which results in feasible individuals. As an example, a *partially mapped crossover* operator is introduced when the encoding of the problem are permutations. Another example of special operators is the restricted search operators, which reduces the search space size, as we will describe later.

Another possibility to enhance the performance of GAs in difficult applications is their hybridization with local search heuristics. There are a wide variety of local search heuristics to be used in hybrid and memetic algorithms, neural networks, simulated annealing, tabu search or just hill-climbing, are some of the local heuristics used herein (Krasnogor & Smith, 2005). When the GA individual is modified after the application of the local search heuristic, the hybrid or memetic algorithm is called *Lamarckian* algorithm[1]. If the local heuristic does not modify the individual, but only its fitness value, it is called a *Baldwin* effect algorithm.

MAIN THRUST

Restricted Search in Genetic Algorithms

Encoding is an important point in genetic algorithms. Many problems in *data mining* require special encodings to be solved by means of GAs. However, the main drawback of not using a traditional binary encoding is that special operators must be used to carry on the crossover and mutation operations. Using binary encodings also have some drawback, mainly the large size of the search space in some problems. In order to reduce the search space size when using GAs with binary encoding, a restricted search operator can be used. Restricted search have been applied in different problems in data mining, such as feature selection (Salcedo-Sanz et al, 2002), (Salcedo-Sanz et al. 2004) or web mining (Salcedo-Sanz & Su, 2006). It consists of an extra operator to be added to the traditional selection, crossover and mutation. This new operator fixes the number of 1s that a given individual can have to a maximum of *p*, reducing the number of 1s if there are more than *p*, and adding 1s at random positions if there are less than *p*. This operation reduces the search space from 2^m to

$$\binom{m}{p},$$

being *m* the length of the binary strings that the GA encodes. Figure 1 shows the outline of the restricted search operator.

Hybrid and Memetic Algorithms

Hybrid algorithms are usually constituted by a genetic algorithm with a local search heuristic to repair infeasible individuals or to calculate the fitness function of the individual. They are specially used in constrained problems, where obtaining feasible individuals randomly is not a trivial task. No impact of the local search in the improvement fitness function value of

Figure 1. Outline of the restricted search operator

The restricted search operator

```
Select m (number of features) before running the GA
for every generation of the GA
    for every individual of the GA population
        check the number of 1s p.
            if(p < m)
                Add_ones(m − p);
            else
                Remove_ones(p − m)
            end(if)
    end(individual)
end(generation)
```

the individual is considered. Usually, the local search heuristic in hybrid genetic algorithms are repairing algorithms, such as Hopfield networks (Salcedo-Sanz et al. 2003), or specific repairing heuristics (Reichelt & Rothlauf, 2005). Other types of neural networks or support vector machines can be embedded if the objective of the local search is to calculate the fitness value of the individual (Sepúlveda-Sanchis, 2002).

The concept of *memetic algorithms* is slightly different. They are also population-based algorithms with a local search, but in this case the local search heuristic has the objective of improving the fitness function exclusively, without reparing anything in the individual.

Examples of memetic algorithms are (Krasnogor & Smith, 2005) (Hart, Krasnogor & Smith, 2005). Usually the local search heuristics used are hill-climbing, tabu search or simulated annealing.

Applications

The restricted search in genetic algorithms has been applied mainly to problems of feature selection and web mining. Salcedo-Sanz et al. introduced this concept in (Salcedo-Sanz et al. 2002) to improve the performance of a wrapper genetic algorithm in the problem of obtaining the best feature selection subset. This application was also the subject of the work (Salcedo-Sanz et al. 2004). An application of the restricted search to inductive query by example can be found in (Salcedo-Sanz & Su, 2006).

Hybrid genetic and memetic algorithms have been also applied to many classification and performance tuning applications in the domain of knowledge discovery in databases (KDD). There are a lot of applications of these approaches to feature selection: (Chakraborty, 2005) proposed a hybrid approach based on the hybridization of neural networks and rough sets, (Li et al. 2005) proposed a hybrid genetic algorithm with a support vector machine as wrapper approach to the problem and (Sabourin, 2005) proposed a multi-objective memetic algorithm for intelligent feature extraction.

Other applications of hybrid and memetic algorithms in data mining are (Buchtala et al., 2005), (Carvalho & Freitas, 2001), (Carvalho & Freitas, 2004), (Carvalho & Freitas, 2002), (Ishibuchi & Yamamoto, 2004), (Freitas, 2001), (Raymer et al. 2003), (Won & Leung, 2004) and in clustering (Cotta & Moscato, 2003), (Cotta & al., 2003), (Merz & Zell, 2003), (Speer et al. 2003), (Vermeulen-Jourdan et al. 2004).

FUTURE TRENDS

The use of hybrid and memetic algorithms is at present a well established trend in data mining applications.

The main drawback of these techniques is, however, the requirement of large computational times, mainly in applications with real databases. The research in hybrid algorithms is focused on developing novel hybrid techniques with low computational time, and still good performance. Also related to this point, it is well known the one of the main drawbacks of applying GAs to data mining is the number of *calls* to the fitness function. Researchers are working for trying to reduce as much as possible the number of scans over the databases in data mining algorithms. To this end, approximation algorithms to the fitness function can be considered, and it is an important line of research in the application of hybrid algorithms to data mining applications.

Also, new heuristic approaches have been discovered in the last few years, such that the *swarm particle optimization* algorithm (Palmer et al. 2005), or the *scattered search* (Glover et al. 2003), which can be successfully hybridized with genetic algorithm in some data mining applications.

CONCLUSION

Hybrid and memetic algorithms are interesting possibilities for enhancing the performance of genetic algorithms in difficult problems. Hybrid algorithms consist of a genetic algorithm with a local search for avoiding infeasible individuals. Also, it can be considered hybrid genetic algorithms those approaches whose fitness calculation involves a specific classifier or regressor. This way, the return of this fitness serves for optimization.

In this chapter, we have revised definitions of memetic algorithms, and how the local search serves to improve GAs individual fitness. Heuristics such as Tabu search, simulated annealing or hill climbing are the most used local search algorithms in memetic approaches. In this chapter, we have studied also other possibilities different from hybrid approaches: special operators for reducing the search space size of genetic algorithms. We have presented one of these possibilities, the restricted search, which has been useful in different applications in the field of feature selection in data mining. Certainly, data mining applications are appealing problems to be tackled with hybrid and memetic algorithms. The election of the encoding, local search heuristic and parameters of the genetic algorithm are the keys to obtain a good approach to this kind of problems.

REFERENCES

Buchtala, O., Klimer, M. Sick, B. (2005). Evolutionary optimization of radial basis function classifiers for data mining applications, *IEEE Transactions on Systems, Man and Cybernetics, part B, 35*(5), 928-947.

Carvalho, D. & Freitas, A. (2001). A genetic algorithm for discovering small-disjunct rules in data mining. *Applied Soft Computing, 2*(2), 75-88.

Carvalho, D. & Freitas, A. (2002). New results for a hybrid decision tree/genetic algorithm for data mining. *Proc. of the 4th Conference on Recent Advances in Soft Computing*, 260-265.

Carvalho, D. & Freitas, A. (2004). A hybrid decision tree/genetic algorithm for data mining. *Information Sciences, 163*, 13-35.

Chakraborty, B. (2005). Feature subset selection by neuro-rough hybridization. *Lecture Notes in Computer Science*, 2005, 519-526.

Cotta, C. & Moscato, P. (2003). A memetic-aided approach to hierarchical clustering from distance matrices: application to gene expression clustering and phylogeny. *Biosystems*, *72*(1), 75-97.

Cotta, C., Mendes, A., García V., Franca, P. & Moscazo, P. (2003). Applying memetic algorithms to the análisis of microarray data. In *Proceedings of the 1st Workshop on Evolutionary Bioinformatics*, Lecture Notes in Computer Science, 2611, 22-32.

Freitas, A. (2001), Understanding the crucial role of attribute interaction in data mining, *Artificial Intelligence Review, 16*(3), 177-199.

Glover, F., Laguna, M. and Martí, R. (2003). *Scatter Search. Advances in evolutionary computation: Techniques and applications.* Springer, 519-537.

Goldberg, D. (1989). *Genetic algorithms in search, optimizacion and machine learning.* Adisson-Wesley.

Hart, W., Krasnogor, N. & Smith, J. (2005) *Recent advances in memetic algorithms*. Springer.

Hordijk, W. & Stadler P. F. (1998). Amplitude spectra of fitness landscapes. *J. Complex Systems, 1*(1):39-66.

Ishibuchi, H. & Yamamoto, T. (2004). Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining, *Fuzzy Sets and Systems, 141*(1), 59-88.

Krasnogor, N. & Smith, J. (2005). A tutorial for competent memetic algorithms: model, taxonomy and design issues, *IEEE Transactions on Evolutionary Computation, 9*(5), 474-488.

Liepins, G. E. & Vose, M. D. (1990). Representational issues in genetic optimization. *J. Experimental and Theoretical Artificial Intelligence, 2*(1):101-115.

Lin, L., Jiang W. & Li, X. (2005). A robust hybrid between genetic algorithm and support vector machines for extracting an optimal feature gene subset, *Genomics, 85*, 16-23.

Merz, P. & Zell, A. (2003). Clustering gene expression profiles with memetic algorithms based on minimum spanning trees, *Lecture Notes in Computer Science, 2439*, 1148-1155.

Palmer, D., Kirschenbaum, M., Shifflet, J. & Seiter, L. (2005). Swarm Reasoning, Proc. of the Swarm Intelligence Symposium, 294-301.

Raymer, M, Doom, T. & Punch, W. (2003). Knowledge discovery in medical and biological data sets using a hybrid Bayes classifier/evolutionary approach, *IEEE Transactions on Systems, Man and Cybernetics, part B, 33*(5), 802-813.

Reichelt, D., & Rothlauf, F. (2005). Reliable communications network design with evolutionary algorithms, *International Journal of Computational Intelligence and Applications, 5*(2), 251-266.

Sabourin, R. (2005). A multi-objective memetic algorithm for intelligent feature extraction, *Lecture Notes in Computer Science, 3410*, 767-775.

Salcedo-Sanz, S. & Su, J. (2006). Improving metaheuristics convergence properties in inductive query by example using two Strategies for reducing the search space, *Computers and Operations Research,* in press.

Salcedo-Sanz, S., Bousoño-Calzón, C. & Figueiras-Vidal, A. (2003), A mixed neural genetic algorithm for the broadcast scheduling problem, *IEEE Transactions on Wireless Communications, 3*(3), 277-283.

Salcedo-Sanz, S., Camps-Valls, G., Pérez-Cruz, F., Sepúlveda-Sanchis, J. & Bousoño-Calzón, C. (2004), Enhancing genetic feature selection though restricted search and walsh analysis, *IEEE Transactions on Systems, Man and Cybernetics, part C, 34*(4), 398-406.

Salcedo-Sanz, S., Prado-Cumplido, M., Pérez-Cruz, F. & Bousoño-Calzón, C. (2002), Feature selection via genetic optimization, *Lecture Notes in Computer Science, 2415*, 547-552.

Sepúlveda, J., Camps-Valls, G., Soria, E. "Support Vector Machines And Genetic Algorithms For detecting Unstable Angina". *The 29th Annual Conference of Computers in Cardiology, CINC'02. IEEE Computer Society Press.* Menphis, USA, September 22-25, 2002

Speer, N., Merz, P., Spieth, C & Zell, A. (2003). Clustering gene expression data with memetic algorithms based on minimum spanning trees, *Proc. of the IEEE Congress on Evolutionary Computation*, 1148-1155.

Vermeulen-Jourdan, L., Dhaenes, C. & Talbi, E., (2004). Clustering nominal and numerical data : a new distance concept for a hybrid genetic algorithm, *Proc. of the 4th European Conference on Evolutionary Computation in Combinatorial Optimization*, 220-229.

Whitley, D.Gordon, V. and Mathias, K. 1994. "Lamarckian Evolution, the Baldwin Effect and Function Optimization", Parallel Problem Solving from Nature-PPSN III, 6-15.

Won, M. & Leung, K. (2004). An efficient data mining method for learning Bayesian networks using an evolutionary algorithm-based hybrid approach, *IEEE Transactions on Evolutionary Computation, 8*(4), 378-404.

KEY TERMS

Baldwin Effect Hybrid Algorithms: In hybrid genetic algorithms, a *Baldwin effect* approach is an algorithm in which the local search does not modify the individual genotype, but only its fitness value.

Fitness Function: In a genetic algorithm, the objective function of the problem is often called *fitness function*, since it measures how well an individual is adapted to its environment (i.e. how good is a given potential solution to the problem we are solving).

Genetic Algorithm: Meta-heuristic technique based on the principles of natural evolution and selection. It is a population based algorithm, where potential solutions (individuals) are evolved through the successive applications of *operators*, such as selection, crossover and mutation.

Hybrid Genetic Algorithms: A kind of global-local search algorithms, formed by a genetic algorithm as global search heuristic, and a local search procedure for ensuring the feasibility of the individuals in the genetic algorithm.

Lamarckian Hybrid Algorithms: In hybrid genetic algorithms, a *Lamarckian* approach is an algorithm in which the local search modifies the individual genotype after its application.

Memetic Algorithm: A hybrid algorithm in which the local search is used as another operator for improving the fitness of the individual.

Simulated Annealing: Another meta-heuristic algorithm also used as local search in memetic and hybrid algorithms. It is inspired by the physical process of heating a substance and then cooling it slowly, until a strong crystalline structure is obtained. This process is simulated by lowering an initial temperature by slow stages until the system reaches to an equilibrium point, and no more changes occur. Each stage of the process consists in changing the configuration several times, until a thermal equilibrium is reached, and a new stage starts, with a lower temperature. The solution of the problem is the configuration obtained in the last stage.

Tabu Search: Meta-heuristic often used as local search in memetic algorithms. It is based on a potential solution to the problem, which is mutated to obtain new solutions from the current one (neighborhood). A *tabu list* is implemented in order to avoid obtaining a solution already visited.

ENDNOTE

[1] The Lamarckian evolution theory states the idea that an organism can acquire characteristics during its life-time and pass them on to its offspring. In the so-called Lamarckian learning the traits acquired during the learning process are passed from parents to their offspring. This means that both the genetic structure of an individual and its associated fitness value are modified to reflect the changes in phenotype structure as a result of performing local search (Whitley et al. 1994).

Imprecise Data and the Data Mining Process

Marvin L. Brown
Grambling State University, USA

John F. Kros
East Carolina University, USA

INTRODUCTION

Missing or inconsistent data has been a pervasive problem in data analysis since the origin of data collection. The management of missing data in organizations has recently been addressed as more firms implement large-scale enterprise resource planning systems (see Vosburg & Kumar, 2001; Xu et al., 2002). The issue of missing data becomes an even more pervasive dilemma in the knowledge discovery process, in that as more data is collected, the higher the likelihood of missing data becomes.

The objective of this research is to discuss imprecise data and the data mining process. The article begins with a background analysis, including a brief review of both seminal and current literature. The main thrust of the chapter focuses on reasons for data inconsistency along with definitions of various types of missing data. Future trends followed by concluding remarks complete the chapter.

BACKGROUND

The analysis of missing data is a comparatively recent discipline. However, the literature holds a number of works that provide perspective on missing data and data mining. Afifi and Elashoff (1966) provide an early seminal paper reviewing the missing data and data mining literature. Little and Rubin's (1987) milestone work defined three unique types of missing data mechanisms and provided parametric methods for handling these types of missing data. These papers sparked numerous works in the area of missing data. Lee and Siau (2001) present an excellent review of data mining techniques within the knowledge discovery process. The references in this section are given as suggested reading for any analyst beginning their research in the area of data mining and missing data.

MAIN THRUST

The article focuses on the reasons for data inconsistency and the types of missing data. In addition, trends regarding missing data and data mining are discussed along with future research opportunities and concluding remarks.

REASONS FOR DATA INCONSISTENCY

Data inconsistency may arise for a number of reasons, including:

- Procedural Factors
- Refusal of Response
- Inapplicable Responses

These three reasons tend to cover the largest areas of missing data in the data mining process.

Procedural Factors

Data entry errors are common and their impact on the knowledge discovery process and data mining can generate serious problems. Inaccurate classifications, erroneous estimates, predictions, and invalid pattern recognition may also take place. In situations where databases are being refreshed with new data, blank responses from questionnaires further complicate the data mining process. If a large number of similar respondents fail to complete similar questions, the deletion or misclassification of these observations can take the researcher down the wrong path of investigation or lead to inaccurate decision-making by end users.

Refusal of Response

Some respondents may find certain survey questions offensive or they may be personally sensitive to certain

questions. For example, some respondents may have no opinion regarding certain questions such as political or religious affiliation. In addition, questions that refer to one's education level, income, age or weight may be deemed too private for some respondents to answer.

Furthermore, respondents may simply have insufficient knowledge to accurately answer particular questions. Students or inexperienced individuals may have insufficient knowledge to answer certain questions (such as salaries in various regions of the country, retirement options, insurance choices, etc).

Inapplicable Responses

Sometimes questions are left blank simply because the questions apply to a more general population rather than to an individual respondent. If a subset of questions on a questionnaire does not apply to the individual respondent, data may be missing for a particular expected group within a data set. For example, adults who have never been married or who are widowed or divorced are likely to not answer a question regarding years of marriage.

TYPES OF MISSING DATA

The following is a list of the standard types of missing data:

- Data Missing at Random
- Data Missing Completely at Random
- Non-Ignorable Missing Data
- Outliers Treated as Missing Data

It is important for an analyst to understand the different types of missing data before they can address the issue. Each type of missing data is defined next.

[Data] Missing at Random (MAR)

Rubin (1978), in a seminal missing data research paper, defined missing data as MAR "when given the variables X and Y, the probability of response depends on X but not on Y." Cases containing incomplete data must be treated differently than cases with complete data. For example, if the likelihood that a respondent will provide his or her weight depends on the probability that the respondent will not provide his or her age, then the missing data is considered to be Missing At Random (MAR) (Kim, 2001).

[Data] Missing Completely at Random (MCAR)

Kim (2001), based on an earlier work, classified data as MCAR when "the probability of response [shows that] independence exists between X and Y." MCAR data exhibits a higher level of randomness than does MAR. In other words, the observed values of Y are truly a random sample for all values of Y, and no other factors included in the study may bias the observed values of Y.

Consider the case of a laboratory providing the results of a chemical compound decomposition test in which a significant level of iron is being sought. If certain levels of iron are met or missing entirely and no other elements in the compound are identified to correlate then it can be determined that the identified or missing data for iron is MCAR.

Non-Ignorable Missing Data

In contrast to the MAR situation where data missingness is explained by other measured variables in a study; non-ignorable missing data arise due to the data missingness pattern being explainable — and only explainable — by the very variable(s) on which the data are missing.

For example, given two variables, X and Y, data is deemed Non-Ignorable when the probability of response depends on variable X and possibly on variable Y. For example, if the likelihood of an individual providing his or her weight varied within various age categories, the missing data is non-ignorable (Kim, 2001). Thus, the pattern of missing data is non-random and possibly predictable from other variables in the database.

In practice, the MCAR assumption is seldom met. Most missing data methods are applied upon the assumption of MAR. And in correspondence to Kim (2001), "Non-Ignorable missing data is the hardest condition to deal with, but unfortunately, the most likely to occur as well."

Outliers Treated as Missing Data

Many times it is necessary to classify these outliers as missing data. Pre-testing and calculating threshold

boundaries are necessary in the pre-processing of data in order to identify those values which are to be classified as missing. Data whose values fall outside of predefined ranges may skew test results. Consider the case of a laboratory providing the results of a chemical compound decomposition test. If it has been predetermined that the maximum amount of iron that can be contained in a particular compound is 500 parts/million, then the value for the variable "iron" should never exceed that amount. If, for some reason, the value does exceed 500 parts/million, then some visualization technique should be implemented to identify that value. Those offending cases are then presented to the end users.

COMMONLY USED METHODS OF ADDRESSING MISSING DATA

Several methods have been developed for the treatment of missing data. The simplest of these methods can be broken down into the following categories:

- Use Of Complete Data Only
- Deleting Selected Cases Or Variables
- Data Imputation

These categories are based on the randomness of the missing data, and how the missing data is estimated and used for replacement. Each category is discussed next.

Use of Complete Data Only

Use of complete data only is generally referred to as the "complete case approach" and is readily available in all statistical analysis packages. When the relationships within a data set are strong enough to not be significantly affected by missing data, large sample sizes may allow for the deletion of a predetermined percentage of cases. While the use of complete data only is a common approach, the cost of lost data and information when cases containing missing value are simply deleted can be dramatic. Overall, this method is best suited to situations where the amount of missing data is small and when missing data is classified as MCAR.

Delete Selected Cases or Variables

The simple deletion of data that contains missing values may be utilized when a non-random pattern of missing data is present. However, it may be ill-advised to eliminate ALL of the samples taken from a test. This method tends to be the method of last choice.

Data Imputation Methods

A number of researchers have discussed specific imputation methods. Seminal works include Little & Rubin (1987), and Rubin (1978) has published articles regarding imputation methodologies. Imputation methods are procedures resulting in the replacement of missing values by attributing them to other available data. This research investigates the most common imputation methods including:

- Case Deletion
- Mean Substitution
- Cold Deck Imputation
- Hot Deck Imputation
- Regression Imputation

These methods were chosen mainly as they are very common in the literature and their ease and expediency of application (Little & Rubin, 1987). However, it has been concluded that although imputation is a flexible method for handling missing-data problems it is not without its drawbacks. Caution should be used when employing imputation methods as they can generate substantial biases between real and imputed data. Nonetheless, imputation methods tend to be a popular method for addressing the issue of missing data.

Case Deletion

The simple deletion of data that contains missing values may be utilized when a nonrandom pattern of missing data is present. Large sample sizes permit the deletion of a predetermined percentage of cases, and/or when the relationships within the data set are strong enough to not be significantly affected by missing data. Case deletion is not recommended for small sample sizes or when the user knows strong relationships within the data exist.

Mean Substitution

This type of imputation is accomplished by estimating missing values by using the mean of the recorded or available values. It is important to calculate the mean only from valid responses that are chosen from a verified population having a normal distribution. If the data distribution is skewed, the median of the available data can be used as a substitute.

The main advantage of mean substitution is its ease of implementation and ability to provide all cases with complete information. Although a common imputation method, three main disadvantages to mean substitution do exist:

- Understatement of variance
- Distortion of actual distribution of values – mean substitution allows more observations to fall into the category containing the calculated mean than may actually exist
- Depression of observed correlations due to the repetition of a constant value

Obviously, a researcher must weigh the advantages against the disadvantages before implementation.

Cold Deck Imputation

Cold deck imputation methods select values or use relationships obtained from sources other than the current data. With this method, the end user substitutes a constant value derived from external sources or from previous research for the missing values. It must be ascertained by the end user that the replacement value used is more valid than any internally derived value. Unfortunately, feasible values are not always provided using cold deck imputation methods. Many of the same disadvantages that apply to the mean substitution method apply to cold deck imputation. Cold deck imputation methods are rarely used as the sole method of imputation and instead are generally used to provide starting values for hot deck imputation methods.

Hot Deck Imputation

Generally speaking, hot deck imputation replaces missing values with values drawn from the next most similar case. The implementation of this imputation method results in the replacement of a missing value with a value selected from an estimated distribution of similar responding units for each missing value. In most instances, the empirical distribution consists of values from responding units. For example, *Table 1* displays a data set containing missing data.

From *Table 1*, it is noted that case three is missing data for item four. In this example, case one, two, and four are examined. Using hot deck imputation, each of the other cases with complete data is examined and the value for the most similar case is substituted for the missing data value. Case four is easily eliminated, as it has nothing in common with case three. Case one and two both have similarities with case three. Case one has one item in common whereas case two has two items in common. Therefore, case two is the most similar to case three.

Once the most similar case has been identified, hot deck imputation substitutes the most similar complete case's value for the missing value. Since case two contains the value of 23 for item four, a value of 23 replaces the missing data point for case three. The advantages of hot deck imputation include conceptual simplicity, maintenance and proper measurement level of variables, and the availability of a complete set of data at the end of the imputation process that can be analyzed like any complete set of data. One of hot deck's disadvantages is the difficulty in defining what is "similar," Hence, many different schemes for deciding on what is "similar" may evolve.

Regression Imputation

Regression Analysis is used to predict missing values based on the variable's relationship to other variables in the data set. Single and/or multiple regression can be used to impute missing values. The first step consists of identifying the independent variables and the

Table 1. Illustration of hot deck imputation: incomplete data set

Case	Item 1	Item 2	Item 3	Item 4
1	10	22	30	25
2	23	20	30	23
3	25	20	30	**???**
4	11	25	10	12

dependent variable. In turn, the dependent variable is regressed on the independent variables. The resulting regression equation is then used to predict the missing values. *Table 2* displays an example of regression imputation.

From the table, twenty cases with three variables (income, age, and years of college education) are listed. Income contains missing data and is identified as the dependent variable while age and years of college education are identified as the independent variables.

The following regression equation is produced for the example

$$\hat{y} = \$79{,}900.95 + \$268.50(\text{age}) + \$2{,}180{,}97(\text{years of college education})$$

Predictions of income can be made using the regression equation and the right-most column of the table displays these predictions. For cases eighteen, nineteen, and twenty, income is predicted to be $107,591.43, $98,728.24, and $101,439.40, respectfully. An advantage to regression imputation is that it preserves the variance and covariance structures of variables with missing data.

Although regression imputation is useful for simple estimates, it has several inherent disadvantages:

- Regression imputation reinforces relationships that already exist within the data – as the method is utilized more often, the resulting data becomes more reflective of the sample and becomes less generalizable
- Understates the variance of the distribution
- An implied assumption that the variable being estimated has a substantial correlation to other attributes within the data set
- Regression imputation estimated value is not constrained and therefore may fall outside predetermined boundaries for the given variable

In addition to these points, there is also the problem of over-prediction. Regression imputation may lead to over-prediction of the model's explanatory power. For example, if the regression R^2 is too strong, multicollinearity most likely exists. Otherwise, if the R^2 value is modest, errors in the regression prediction equation will be substantial. Overall, regression imputation not only estimates the missing values but also derives inferences for the population.

Table 2. Illustration of regression imputation

Case	Income	Age	Years of College Education	Regression Prediction
1	$95,131.25	26	4	$96,147.60
2	$108,664.75	45	6	$104,724.04
3	$98,356.67	28	5	$98,285.28
4	$94,420.33	28	4	$96,721.07
5	$104,432.04	46	3	$100,318.15
6	$97,151.45	38	4	$99,588.46
7	$98,425.85	35	4	$98,728.24
8	$109,262.12	50	6	$106,157.73
9	$95,704.49	45	3	$100,031.42
10	$99,574.75	52	5	$105,167.00
11	$96,751.11	30	0	$91,037.71
12	$111,238.13	50	6	$106,157.73
13	$102,386.59	46	6	$105,010.78
14	$109,378.14	48	6	$105,584.26
15	$98,573.56	50	4	$103,029.32
16	$94,446.04	31	3	$96,017.08
17	$101,837.93	50	4	$103,029.32
18	???	55	6	$107,591.43
19	???	35	4	$98,728.24
20	???	39	5	$101,439.40

FUTURE TRENDS

Poor data quality has plagued the knowledge discovery process and all associated data mining techniques. Future data mining systems should be sensitive to noise and have the ability to deal with all types of data pollution, both internally and in conjunction with end users. Systems should still produce the most significant findings from the data set possible even if noise is present.

As data mining continues to evolve and mature as a viable business tool, it will be monitored to address its role in the technological life cycle. Tools for dealing with missing data will grow from being used as a horizontal solution (not designed to provide business-specific end solutions) into a type of vertical solution (integration of domain-specific logic into data mining

solutions). As gigabyte-, terabyte-, and petabyte-size data sets become more prevalent in data warehousing applications, the issue of dealing with missing data will itself become an integral solution for the use of such data rather than simply existing as a component of the knowledge discovery and data mining processes (Han & Kamber, 2001).

Although the issue of statistical analysis with missing data has been addressed since the early 1970s, the advent of data warehousing, knowledge discovery, data mining and data cleansing has pushed the concept of dealing with missing data into the limelight. Brown & Kros (2003) provide an overview of the trends, techniques, and impacts of imprecise data on the data mining process related to the *k*-Nearest Neighbor Algorithm, Decision Trees, Association Rules, and Neural Networks.

CONCLUSION

It can be seen that future generations of data miners will be faced with many challenges concerning the issues of missing data. This article gives a background analysis and brief literature review of missing data concepts. The authors addressed reasons for data inconsistency and methods for addressing missing data. Finally, the authors offered their opinions on future developments and trends on issues expected to face developers of knowledge discovery software and the needs of end users when confronted with the issues of data inconsistency and missing data.

REFERENCES

Afifi, A., & Elashoff, R. (1966). Missing observations in multivariate statistics I: Review of the literature. *Journal of the American Statistical Association, 61*, 595-604.

Brown, M.L., & Kros, J.F. (2003). The impact of missing data on data mining. In J. Wang (Ed.), *Data mining opportunities and challenges* (pp. 174-198). Hershey, PA: Idea Group Publishing.

Han, J., & Kamber, M. (2001). *Data mining: Concepts and techniques*. San Francisco: Academic Press.

Kim, Y. (2001). *The curse of the missing data*. Retrieved from *http://209.68.240.11:8080/2ndMoment/978476655/addPostingForm/*

Lee, S., & Siau, K. (2001). A review of data mining techniques. *Industrial Management & Data Systems*, *101*(1), 41-46.

Little, R., & Rubin, D. (1987). *Statistical analysis with missing data*. New York: Wiley.

Rubin, D. (1978). Multiple imputations in sample surveys: A phenomenological Bayesian approach to nonresponse. In *Imputation and Editing of Faulty or Missing Survey Data* (pp. 1-23). Washington, DC: U.S. Department of Commerce.

Vosburg, J., & Kumar, A. (2001). Managing dirty data in organizations using ERP: Lessons from a case study. *Industrial Management & Data Systems*, *101*(1), 21-31.

Xu, H., Horn Nord, J., Brown, N., & Nord, G.D. (2002). Data quality issues in implementing an ERP. *Industrial Management & Data Systems, 102*(1), 47-58.

KEY TERMS

[Data] Missing Completely at Random (MCAR): When the observed values of a variable are truly a random sample of all values of that variable (i.e., the response exhibits independence from any variables).

Data Imputation: The process of estimating missing data of an observation based on the valid values of other variables.

Data Missing at Random (MAR): When given the variables X and Y, the probability of response depends on X but not on Y.

Inapplicable Responses: Respondents omit answer due to doubts of applicability.

Knowledge Discovery Process: The overall process of information discovery in large volumes of warehoused data.

Non-Ignorable Missing Data: Arise due to the data missingness pattern being explainable, non-random, and possibly predictable from other variables.

Procedural Factors: Inaccurate classifications of new data, resulting in classification error or omission.

Refusal of Response: Respondents outward omission of a response due to personal choice, conflict, or inexperience.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 593-598, copyright 2005 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

Incremental Learning

Abdelhamid Bouchachia
University of Klagenfurt, Austria

INTRODUCTION

Data mining and knowledge discovery is about creating a comprehensible model of the data. Such a model may take different forms going from simple association rules to complex reasoning system. One of the fundamental aspects this model has to fulfill is adaptivity. This aspect aims at making the process of knowledge extraction continually maintainable and subject to future update as new data become available. We refer to this process as *knowledge learning*.

Knowledge learning systems are traditionally built from data samples in an off-line one-shot experiment. Once the learning phase is exhausted, the learning system is no longer capable of learning further knowledge from new data nor is it able to update itself in the future. In this chapter, we consider the problem of incremental learning (IL). We show how, in contrast to off-line or batch learning, IL learns knowledge, be it symbolic (e.g., rules) or sub-symbolic (e.g., numerical values) from data that evolves over time. The basic idea motivating IL is that as new data points arrive, new knowledge elements may be created and existing ones may be modified allowing the knowledge base (respectively, the system) to evolve over time. Thus, the acquired knowledge becomes self-corrective in light of new evidence. This update is of paramount importance to ensure the adaptivity of the system. However, it should be meaningful (by capturing only interesting events brought by the arriving data) and sensitive (by safely ignoring unimportant events). Perceptually, IL is a fundamental problem of cognitive development. Indeed, the perceiver usually learns how to make sense of its sensory inputs in an incremental manner via a filtering procedure.

In this chapter, we will outline the background of IL from different perspectives: machine learning and data mining before highlighting our IL research, the challenges, and the future trends of IL.

BACKGROUND

IL is a key issue in applications where data arrives over long periods of time and/or where storage capacities are very limited. Most of the knowledge learning literature reports on learning models that are one-shot experience. Once the learning stage is exhausted, the induced knowledge is no more updated. Thus, the performance of the system depends heavily on the data used during the learning (knowledge extraction) phase. Shifts of trends in the arriving data cannot be accounted for.

Algorithms with an IL ability are of increasing importance in many innovative applications, e.g., video streams, stock market indexes, intelligent agents, user profile learning, etc. Hence, there is a need to devise learning mechanisms that are able of accommodating new data in an incremental way, while keeping the system under use. Such a problem has been studied in the framework of adaptive resonance theory (Carpenter et al., 1991). This theory has been proposed to efficiently deal with the stability-plasticity dilemma. Formally, a learning algorithm is totally stable if it keeps the acquired knowledge in memory without any catastrophic forgetting. However, it is not required to accommodate new knowledge. On the contrary, a learning algorithm is completely plastic if it is able to continually learn new knowledge without any requirement on preserving the knowledge previously learned. The dilemma aims at accommodating new data (plasticity) without forgetting (stability) by generating knowledge elements over time whenever the new data conveys new knowledge elements worth considering.

Basically there are two schemes to accommodate new data. To retrain the algorithm from scratch using both old and new data is known as revolutionary strategy. In contrast, an evolutionary continues to train the algorithm using only the new data (Michalski, 1985). The first scheme fulfills only the stability requirement, whereas the second is a typical IL scheme that is able to fulfill both stability and plasticity. The goal is to make

a tradeoff between the stability and plasticity ends of the learning spectrum as shown in Figure 1.

As noted in (Polikar et al., 2000), there are many approaches referring to some aspects of IL. They exist under different names like *on-line learning*, *constructive learning*, *lifelong learning*, and *evolutionary learning*. Therefore, a definition of IL turns out to be vital:

- IL should be able to accommodate plasticity by learning knowledge from new data. This data can refer either to the already known structure or to a new structure of the system.
- IL can use only new data and should not have access at any time to the previously used data to update the existing system.
- IL should be able to observe the stability of the system by avoiding forgetting.

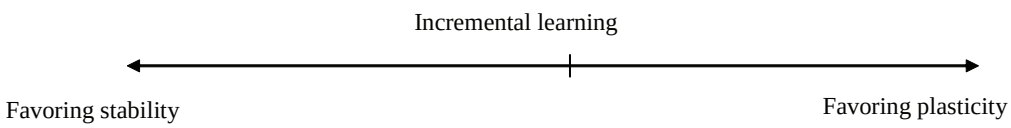
It is worth noting that the IL research flows in three directions: clustering, classification, and rule associations mining. In the context of classifcation and clustering, many IL approaches have been introduced. A typical incremental approach is discussed in (Parikh & Polikar, 2007) which consists of combining an ensemble of multilayer perceptron networks (MLP) to accommodate new data. Note that stand-alone MLPs, like many other neural networks, need retraining in order to learn from the new data. Other IL algorithms were also proposed as in (Domeniconi & Gunopulos, 2001), where the aim is to construct incremental support vector machine classifiers. Actually, there exist four neural models that are inherently incremental: (i) adaptive resonance theory (ART) (Carpenter et al., 1991), (ii) min-max neural networks (Simpson, 1992), (iii) nearest generalized exemplar (Salzberg, 1991), and (iv) neural gas model (Fritzke, 1995). The first three incremental models aim at learning hyper-rectangle categories, while the last one aims at building point-prototyped categories.

It is important to mention that there exist many classification approaches that are referred to as IL approaches and which rely on neural networks. These range from retraining misclassified samples to various weighing schemes (Freeman & Saad, 1997; Grippo, 2000). All of them are about sequential learning where input samples are sequentially, but iteratively, presented to the algorithm. However, sequential learning works only in close-ended environments where classes to be learned have to be reflected by the readily available training data and more important prior knowledge can also be forgotten if the classes are unbalanced.

In contrast to sub-symbolic learning, few authors have studied incremental symbolic learning, where the problem is incrementally learning simple classification rules (Maloof & Michalski, 2004; Reinke & Michalski, 1988).

In addition, the concept of incrementality has been discussed in the context of association rules mining (ARM). The goal of ARM is to generate all association rules in the form of $X \rightarrow Y$ that have support and confidence greater than a user-specified minimum support and minimum confidence respectively. The motivation underlying incremental ARM stems from the fact that databases grow over time. The association rules mined need to be updated as new items are inserted in the database. Incremental ARM aims at using only the incremental part to infer new rules. However, this is usually done by processing the incremental part separately and scanning the older database if necessary. Some of the algorithms proposed are FUP (Cheung et al., 1996), temporal windowing (Rainsford et al., 1997), and DELI (Lee & Cheung, 1997).

Figure 1. Learning spectrum



In contrast to static databases, IL is more visible in data stream ARM. The nature of data imposes such an incremental treatment of data. Usually data continually arrives in the form of high-speed streams. IL is particularly relevant for online streams since data is discarded as soon as it has been processed. Many algorithms have been introduced to maintain association rules (Charikar et al., 2004; Gama et al., 2007). Furthermore, many classification clustering algorithm, which are not fully incremental, have been developed in the context of stream data (Aggarwal et al., 2004; Last, 2002).

FOCUS

IL has a large spectrum of investigation facets. We shall focus in the following on classification and clustering which are key issues in many domains such as data mining, pattern recognition, knowledge discovery, and machine learning. In particular, we focus on two research avenues which we have investigated: (i) incremental fuzzy classifiers (IFC) (Bouchachia & Mittermeir, 2006) and (ii) incremental learning by function decomposition (IFD) (Bouchachia, 2006a).

The motivation behind IFC is to infer knowledge in the form of fuzzy rules from data that evolves over time. To accommodate IL, appropriate mechanisms are applied in all steps of the fuzzy system construction:

1. **Incremental supervised clustering:** Given a labeled data set, the first step is to cluster this data with the aim of achieving high purity and separability of clusters. To do that, we have introduced a clustering algorithm that is *incremental* and *supervised*. These two characteristics are vital for the whole process. The resulting labeled clusters' prototypes are projected onto each feature axis to generate some fuzzy partitions.
2. **Fuzzy partitioning and accommodation of change:** Fuzzy partitions are generated relying on two steps: Initially, each cluster is mapped onto a triangular partition. In order to optimize the shape of the partitions, the number and the complexity of rules, an aggregation of these triangular partitions is performed. As new data arrives, these partitions are systematically updated without referring to the previously used data. The consequent of rules are then accordingly updated.
3. **Incremental feature selection:** To find the most relevant features (which results in compact and transparent rules), an *incremental version* of Fisher's interclass separability criterion is devised. As new data arrives, some features may be substituted for new ones in the rules. Hence, the rules' premises are dynamically updated. At any time of the life of a classifier, the rule base should reflect the semantic contents of the already used data. To the best of our knowledge, there is no previous work on feature selection algorithms that observe the notion of *incrementality*.

In another research axis, IL has been thoroughly investigated in the context of neural networks. In (Bouchachia, 2006a) we have proposed a novel IL algorithm based on function decomposition (ILFD) that is realized by a neural network. ILFD uses clustering and vector quantization techniques to deal with classification tasks. The main motivation behind ILFD is to enable an on-line classification of data lying in different regions of the space allowing to generate non-convex partitions and, more generally, to generate disconnected partitions (not lying in the same contiguous space). Hence, each class can be approximated by a sufficient number of categories centered around their prototypes. Furthermore, ILFD differs from the aforementioned learning techniques (Sec. Background) with respect to the following aspects:

- Most of those techniques rely on geometric shapes to represent the categories, such as hyper-rectangles, hyper-ellipses, etc.; whereas the ILFD approach is not explicitly based on a particular shape since one can use different types of distances to obtain different shapes.
- Usually, there is no explicit mechanism (except for the neural gas model) to deal with redundant and dead categories, the ILFD approach uses two procedures to get rid of dead categories. The first is called *dispersion test* and aims at eliminating redundant category nodes. The second is called *staleness test* and aims at pruning categories that become stale.
- While all of those techniques modify the position of the winner when presenting the network with a data vector, the learning mechanism in ILFD consists of reinforcing the winning category from the class of the data vector and pushes away the

second winner from a neighboring class to reduce the overlap between categories.
- While the other approaches are either self-supervised or need to match the input with all existing categories, ILFD compares the input only with categories having the same label as the input in the first place and then with categories from other labels distinctively.
- The ILFD can also deal with the problem of partially labeled data. Indeed, even unlabeled data can be used during the training stage.

Moreover, the characteristics of ILFD can be compared to other models such as fuzzy ARTMAP (FAM), min-max neural networks (MMNN), nearest generalized exemplar (NGE), and growing neural gas (GNG) as shown in Table 1 (Bouchachia et al., 2007).

In our research, we have tried to stick to the spirit of IL. To put it clearly, an IL algorithm, in our view, should fulfill the following characteristics:

- Ability of life-long learning and to deal with plasticity and stability
- Old data is never used in subsequent stages
- No prior knowledge about the (topological) structure of the system is needed
- Ability to incrementally tune the structure of the system
- No prior knowledge about the statistical properties of the data is needed
- No prior knowledge about the number of existing classes and the number of categories per class and no prototype initialization are required.

FUTURE TRENDS

The problem of incrementality remains a key aspect in learning systems. The goal is to achieve adaptive systems that are equipped with self-correction and evolution mechanisms. However, many issues, which can be seen as shortcomings of existing IL algorithms, remain open and therefore worth investigating:

- **Order of data presentation:** All of the proposed IL algorithms suffer from the problem of sensitivity to the order of data presentation. Usually, the inferred classifiers are biased by this order. Indeed

Table 1. Characteristics of some IL algorithms

Characteristics	FAM	MMNN	NGE	GNG	ILFD
Online learning	Y	Y	Y	Y	Y
Type of prototypes	Hyperbox	Hyperbox	Hyperbox	Graph node	Cluster center
Generation control	Y	Y	Y	Y	Y
Shrinking of prototypes	N	Y	Y	U	U
Deletion of prototypes	N	N	N	Y	Y
Overlap of prototypes	Y	N	N	U	U
Growing of prototypes	Y	Y	Y	U	U
Noise resistance	U	Y	U	U	U
Sensitivity to data order	Y	Y	Y	Y	Y
Normalization	Y	Y	Y/N	N	Y/N

Legend: Y: yes N: no U: uknown/undefined

different presentation orders result in different classifier structures and therefore in different accuracy levels. It is therefore very relevant to look closely at developing algorithms whose behavior is data-presentation independent. Usually, this is a desired property.

- **Category proliferation:** The problem of category proliferation in the context of clustering and classification refers to the problem of generating a large number of categories. This number is in general proportional to the granularity of categories. In other terms, fine category size implies large number of categories and larger size implies less categories. Usually, there is a parameter in each IL algorithm that controls the process of category generation. The problem here is: what is the appropriate value of such a parameter. This is clearly related to the problem of plasticity that plays a central role in IL algorithms. Hence, the question: how can we distinguish between rare events and outliers? What is the controlling parameter value that allows making such a distinction? This remains a difficult issue.
- **Number of parameters:** One of the most important shortcomings of the majority of the IL algorithms is the huge number of user-specified parameters that are involved. It is usually hard to find the optimal value of these parameters. Furthermore, they are very sensitive to data, i.e., in general to obtain high accuracy values, the setting requires change from one data set to another. In this context, there is a real need to develop algorithms that do not depend heavily on many parameters or which can optimize such parameters.
- **Self-consciousness & self-correction:** The problem of distinction between noisy input data and rare event is not only crucial for category generation, but it is also for correction. In the current approaches, IL systems cannot correct wrong decisions made previously, because each sample is treated once and any decision about it has to be taken at that time. Now, assume that at the processing time the sample *x* was considered a noise, while in reality it was a rare event, then in a later stage the same rare event was discovered by the system. Therefore, in the ideal case the system has to recall that the sample *x* has to be reconsidered. Current algorithms are not able to adjust the systems by re-examining old decisions. Thus, IL systems have to be equipped with some memory in order to become smarter enough.
- **Data drift:** One of the most difficult questions that is worth looking at is related to drift. Little, if none, attention has been paid to the application and evaluation of the aforementioned IL algorithms in the context of drifting data although the change of environment is one of the crucial assumptions of all these algorithms. Furthermore, there are many publicly available datasets for testing systems within static setting, but there are very few benchmark data sets for dynamically changing problems. Those existing are usually artificial sets. It is very important for the IL community to have a repository, similar to that of the Irvine UCI, in order to evaluate the proposed algorithms in evolving environments.

As a final aim, the research in the IL framework has to focus on incremental but stable algorithms that have to be transparent, self-corrective, less sensitive to the order of data arrival, and whose parameters are less sensitive to the data itself.

CONCLUSION

Building adaptive systems that are able to deal with nonstandard settings of learning is one of key research avenues in machine learning, data mining and knowledge discovery. Adaptivity can take different forms, but the most important one is certainly incrementality. Such systems are continuously updated as more data becomes available over time. The appealing features of IL, if taken into account, will help integrate intelligence into knowledge learning systems. In this chapter we have tried to outline the current state of the art in this research area and to show the main problems that remain unsolved and require further investigations.

REFERENCES

Aggarwal, C., Han, J., Wang, J., & Yu, P. (2004). On demand classification of data streams. *International Conference on Knowledge Discovery and Data Mining*, pp. 503-508.

Bouchachia, A. & Mittermeir, R. (2006). Towards fuzzy incremental classifiers. *Soft Computing*, *11*(2), 193-207.

Bouchachia, A., Gabrys, B. & Sahel, Z. (2007). Overview of some incremental learning algorithms. *Proc. of the 16th IEEE international conference on fuzzy systems*, IEEE Computer Society, 2007

Bouchachia, A. (2006a). Learning with incrementality. *The 13th International conference on neural information processing*, LNCS 4232, pp. 137-146.

Carpenter, G., Grossberg, D., & Rosen, D. (1991). Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks, 4*(6), 759–771.

Charikar, M., Chen, K., & Farach-Colton, M. (2004). Finding frequent items in data streams. *International Colloquium on Automata, Languages and Programming*, pp. 693–703.

Cheung, D., Han, J., Ng, V., & Wong, C. (1996). Maintenance of discovered association rules in large databases: An incremental updating technique. *IEEE International Conference on Data Mining*, 106-114.

Domeniconi, C. & Gunopulos, D. (2001). Incremental Support Vector Machine Construction. *International Conference on Data Mining*, pp. 589-592.

Freeman, J. & Saad, D. (1997). On-line learning in radial basis function networks. *Neural Computation, 9*, 1601-1622.

Fritzke, B. (1995). A growing neural gas network learns topologies. *Advances in neural information processing systems*, pp. 625-632.

Grippo, L. (2000). Convergent on-line algorithms for supervised learning in neural networks. *IEEE Trans. on Neural Networks*, *11*, 1284–1299.

Gama, J. Rodrigues, P., and Aguilar-Ruiz, J. (2007). An Overview on learning from data streams. *New Generation Computing*, *25*(1), 1-4.

Last, M. (2002). Online classification of non-stationary data streams. *Intelligent Data Analysis*, *6*(2), 129-147.

Lee, S., & Cheung, D. (1997). Maintenance of discovered association rules: when to update?. *SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery.*

Maloof, M., & Michalski, R. (2004). Incremental learning with partial instance memory. *Artificial Intelligence 154*, 95–126.

Michalski, R. (1985). Knowledge repair mechanisms: evolution vs. revolution. *International Machine Learning Workshop*, pp. 116–119.

Parikh, D. & Polikar, R. (2007). An ensemble-based incremental learning approach to data fusion. *IEEE transaction on Systems, Man and Cybernetics*, *37*(2), 437-450.

Polikar, R., Udpa, L., Udpa, S. & Honavar, V. (2000). Learn++: An incremental learning algorithm for supervised neural networks. *IEEE Trans. on Systems, Man, and Cybernetics*, *31*(4), 497–508.

Rainsford, C., Mohania, M., & Roddick, J. (1997). A temporal windowing approach to the incremental maintenance of association rules. *International Database Workshop, Data Mining, Data Warehousing and Client/Server Databases*, pp. 78-94.

Reinke, R., & Michalski, R. (1988). *Machine intelligence*, chapter: Incremental learning of concept descriptions: a method and experimental results, pp. 263–288.

Salzberg, S. (1991). A nearest hyperrectangle learning method. *Machine learning*, *6*, 277–309.

Simpson, P. (1992). Fuzzy min-max neural networks. Part 1: Classification. *IEEE Trans. Neural Networks*, *3*(5), 776-786.

KEY TERMS

Data Drift: Unexpected change over time of the data values (according to one or more dimensions).

Incrementality: The characteristic of an algorithm that is capable of processing data which arrives over time sequentially in a stepwise manner without referring to the previously seen data.

Knowledge Learning: Knowledge Learning: The process of automatic extracting Knowledge from data.

Plasticity: A learning algorithm is completely plastic if it is able to continually learn new knowledge without any requirement on preserving previously seen data.

Stability: A learning algorithm is totally stable if it keeps the acquired knowledge in memory without any catastrophic forgetting.

Incremental Mining from News Streams

Seokkyung Chung
University of Southern California, USA

Jongeun Jun
University of Southern California, USA

Dennis McLeod
University of Southern California, USA

INTRODUCTION

With the rapid growth of the World Wide Web, Internet users are now experiencing overwhelming quantities of online information. Since manually analyzing the data becomes nearly impossible, the analysis would be performed by automatic data mining techniques to fulfill users' information needs quickly.

On most Web pages, vast amounts of useful knowledge are embedded into text. Given such large sizes of text collection, mining tools, which organize the text datasets into structured knowledge, would enhance efficient document access. This facilitates information search and, at the same time, provides an efficient framework for document repository management as the number of documents becomes extremely huge.

Given that the Web has become a vehicle for the distribution of information, many news organizations are providing newswire services through the Internet. Given this popularity of the Web news services, text mining on news datasets has received significant attentions during the past few years. In particular, as several hundred news stories are published everyday at a single Web news site, triggering the whole mining process whenever a document is added to the database is computationally impractical. Therefore, efficient incremental text mining tools need to be developed.

BACKGROUND

The simplest document access method within Web news services is keyword-based retrieval. Although this method seems effective, there exist at least three serious drawbacks. First, if a user chooses irrelevant keywords, then retrieval accuracy will be degraded. Second, since keyword-based retrieval relies on the syntactic properties of information (e.g., keyword counting), *semantic gap* cannot be overcome (Grosky, Sreenath, & Fotouhi, 2002). Third, only expected information can be retrieved since the specified keywords are generated from users' knowledge space. Thus, if users are unaware of the airplane crash that occurred yesterday, then they cannot issue a query about that accident even though they might be interested.

The first two drawbacks stated above have been addressed by query expansion based on domain-independent ontologies. However, it is well known that this approach leads to a degradation of precision. That is, given that the words introduced by term expansion may have more than one meaning, using additional terms can improve recall, but decrease precision. Exploiting a manually developed ontology with a controlled vocabulary would be helpful in this situation (Khan, McLeod, & Hovy, 2004). However, although ontology-authoring tools have been developed in the past decades, manually constructing ontologies whenever new domains are encountered is an error-prone and time-consuming process. Therefore, integration of knowledge acquisition with data mining, which is referred to as *ontology learning*, becomes a must (Maedche & Staab, 2001).

To facilitate information navigation and search on a news database, clustering can be utilized. Since a collection of documents is easy to skim if similar articles are grouped together, if the news articles are hierarchically classified according to their topics, then a query can be formulated while a user navigates a cluster hierarchy. Moreover, clustering can be used to identify and deal with near-duplicate articles. That is, when news feeds repeat stories with minor changes from hour to hour, presenting only the most recent articles is probably sufficient. In particular, a sophisticated incremental

hierarchical document clustering algorithm can be effectively used to address high rate of document update. Moreover, in order to achieve rich semantic information retrieval, an ontology-based approach would be provided. However, one of the main problems with concept-based ontologies is that topically related concepts and terms are not explicitly linked. That is, there is no relation between *court-attorney*, *kidnap-police*, and etcetera. Thus, concept-based ontologies have a limitation in supporting a topical search. In sum, it is essential to develop incremental text mining methods for intelligent news information presentation.

MAIN THRUST

In the following, we will explore text mining approaches that are relevant for news streams data.

Requirements of Document Clustering in News Streams

Data we are considering are high dimensional, large in size, noisy, and a continuous stream of documents. Many previously proposed document clustering algorithms did not perform well on this dataset due to a variety of reasons. In the following, we define application-dependent (in terms of news streams) constraints that the clustering algorithm must satisfy.

1. **Ability to determine input parameters:** Many clustering algorithms require a user to provide input parameters (e.g., the number of clusters), which is difficult to be determined in advance, in particular when we are dealing with incremental datasets. Thus, we expect the clustering algorithm not to need such kind of knowledge.
2. **Scalability with large number of documents:** The number of documents to be processed is extremely large. In general, the problem of clustering *n* objects into *k* clusters is NP-hard. Successful clustering algorithms should be scalable with the number of documents.
3. **Ability to discover clusters with different shapes and sizes:** The shape of document cluster can be of arbitrary shapes; hence we cannot assume the shape of document cluster (e.g., hyper-sphere in *k*-means). In addition, the sizes of clusters can be of arbitrary numbers, thus clustering algorithms should identify the clusters with wide variance in size.
4. **Outliers Identification:** In news streams, outliers have a significant importance. For instance, a unique document in a news stream may imply a new technology or event that has not been mentioned in previous articles. Thus, forming a singleton cluster for the outlier is important.
5. **Efficient incremental clustering:** Given different ordering of a same dataset, many incremental clustering algorithms produce different clusters, which is an unreliable phenomenon. Thus, the incremental clustering should be robust to the input sequence. Moreover, due to the frequent document insertion into the database, whenever a new document is inserted it should perform a fast update of the existing cluster structure.
6. **Meaningful theme of clusters:** We expect each cluster to reflect a meaningful theme. We define "meaningful theme" in terms of precision and recall. That is, if a cluster (*C*) is about "Turkey earthquake,'' then all documents about "Turkey earthquake'' should belong to *C*, and documents that do not talk about "Turkey earthquake" should not belong to *C*.
7. **Interpretability of resulting clusters:** A clustering structure needs to be tied up with a succinct summary of each cluster. Consequently, clustering results should be easily comprehensible by users.

Previous Document Clustering Approaches

The most widely used document clustering algorithms fall into two categories: partition-based clustering and hierarchical clustering. In the following, we provide a concise overview for each of them, and discuss why these approaches fail to address the requirements discussed above.

Partition-based clustering decomposes a collection of documents, which is optimal with respect to some pre-defined function (Duda, Hart, & Stork, 2001; Liu, Gong, Xu, & Zhu, 2002). Typical methods in this category include center-based clustering, Gaussian Mixture Model, and etcetera. Center-based algorithms identify the clusters by partitioning the entire dataset into a pre-determined number of clusters (e.g., *k*-means clustering). Although the center-based clustering algo-

rithms have been widely used in document clustering, there exist at least five serious drawbacks. First, in many center-based clustering algorithms, the number of clusters needs to be determined beforehand. Second, the algorithm is sensitive to an initial seed selection. Third, it can model only a spherical (*k*-means) or ellipsoidal (*k*-medoid) shape of clusters. Furthermore, it is sensitive to outliers since a small amount of outliers can substantially influence the mean value. Note that capturing an outlier document and forming a singleton cluster is important. Finally, due to the nature of an iterative scheme in producing clustering results, it is not relevant for incremental datasets.

Hierarchical (agglomerative) clustering (HAC) identifies the clusters by initially assigning each document to its own cluster and then repeatedly merging pairs of clusters until a certain stopping condition is met (Zhao & Karypis, 2002). Consequently, its result is in the form of a tree, which is referred to as a *dendrogram*. A dendrogram is represented as a tree with numeric levels associated to its branches. The main advantage of HAC lies in its ability to provide a view of data at multiple levels of abstraction. Although HAC can model arbitrary shapes and different sizes of clusters, and can be extended to the robust version (in outlier handling sense), it is not relevant for news streams application due to the following two reasons. First, since HAC builds a dendrogram, a user should determine where to cut the dendrogram to produce actual clusters. This step is usually done by human visual inspection, which is a time-consuming and subjective process. Second, the computational complexity of HAC is expensive since pairwise similarities between clusters need to be computed.

Topic Detection and Tracking

Over the past six years, the information retrieval community has developed a new research area, called TDT (Topic Detection and Tracking) (Makkonen, Ahonen-Myka, & Salmenkivi, 2004; Allan, 2002). The main goal of TDT is to detect the occurrence of a novel event in a stream of news stories, and to track the known event. In particular, there are three major components in TDT.

1. **Story segmentation:** It segments a news stream (e.g., including transcribed speech) into topically cohesive stories. Since online Web news (in HTML format) is supplied in segmented form, this task only applies to audio or TV news.
2. **First Story Detection (FSD):** It identifies whether a new document belongs to an existing topic or a new topic.
3. **Topic tracking:** It tracks events of interest based on sample news stories. It associates incoming news stories with the related stories, which were already discussed before. It can be also asked to monitor the news stream for further stories on the same topic.

Event is defined as "some unique thing that happens at some point in time". Hence, an event is different from a topic. For example, "airplane crash" is a topic while "Chinese airplane crash in Korea in April 2002" is an event. Note that it is important to identify events as well as topics. Although a user is not interested in a flood topic, the user may be interested in the news story on the Texas flood if the user's hometown is from Texas. Thus, a news recommendation system must be able to distinguish different events within a same topic.

Single-pass document clustering (Chung & McLeod, 2003) has been extensively used in TDT research. However, the major drawback of this approach lies in order-sensitive property. Although the order of documents is already fixed since documents are inserted into the database in chronological order, order-sensitive property implies that the resulting cluster is unreliable. Thus, new methodology is required in terms of incremental news article clustering.

Dynamic Topic Mining

Dynamic topic mining is a framework that supports the identification of meaningful patterns (e.g., events, topics, and topical relations) from news stream data (Chung & McLeod, 2003). To build a novel paradigm for an intelligent news database management and navigation scheme, it utilizes techniques in information retrieval, data mining, machine learning, and natural language processing.

In dynamic topic mining, a Web crawler downloads news articles from a news Web site on a daily basis. Retrieved news articles are processed by diverse information retrieval and data mining tools to produce useful higher-level knowledge, which is stored in a content description database. Instead of interacting with a Web news service directly, by exploiting the knowl-

edge in the database, an information delivery agent can present an answer in response to a user request (in terms of topic detection and tracking, keyword-based retrieval, document cluster visualization, etc). Key contributions of the dynamic topic mining framework are development of a novel hierarchical incremental document clustering algorithm, and a topic ontology learning framework.

Despite the huge body of research efforts on document clustering, previously proposed document clustering algorithms are limited in that it cannot address special requirements in a news environment. That is, an algorithm must address the seven application-dependent constraints discussed before. Toward this end, the dynamic topic mining framework presents a sophisticated incremental hierarchical document clustering algorithm that utilizes a neighborhood search. The algorithm was tested to demonstrate the effectiveness in terms of the seven constraints. The novelty of the algorithm is the ability to identify meaningful patterns (e.g., news events, and news topics) while reducing the amount of computations by maintaining cluster structure incrementally.

In addition, to overcome the lack of topical relations in conceptual ontologies, a topic ontology learning framework is presented. The proposed topic ontologies provide interpretations of news topics at different levels of abstraction. For example, regarding to a Winona Ryder court trial news topic (*T*), the dynamic topic mining could capture "winona, ryder, actress, shoplift, beverly" as specific terms describing *T* (i.e., the specific concept for *T*) while "attorney, court, defense, evidence, jury, kill, law, legal, murder, prosecutor, testify, trial" as general terms representing *T* (i.e., the general concept for *T*). There exists research work on extracting hierarchical relations between terms from a set of documents (Tseng, 2002). However, the dynamic topic mining framework is unique in that the topical relations are dynamically generated based on incremental hierarchical clustering rather than based on human defined topics, such as Yahoo directory.

FUTURE TRENDS

There are many future research opportunities in news streams mining. First, although a document hierarchy can be obtained using unsupervised clustering, as shown in Aggarwal, Gates, & Yu (2004), the cluster quality can be enhanced if a pre-existing knowledge base is exploited. That is, based on this priori knowledge, we can have some control while building a document hierarchy.

Second, document representation for clustering can be augmented with phrases by employing different levels of linguistic analysis (Hatzivassiloglou, Gravano, & Maganti, 2000). That is, representation model can be augmented by adding *n*-gram (Peng & Schuurmans, 2003), or frequent itemsets using association rule mining (Hand, Mannila, & Smyth, 2001). Investigating how different feature selection algorithms affect on the accuracy of clustering results is an interesting research work.

Third, besides exploiting text data, other information can be utilized since Web news articles are composed of text, hyperlinks, and multimedia data. For example, both terms and hyperlinks (which point to related news articles or Web pages) can be used for feature selection.

Finally, a topic ontology learning framework can be extended to accommodating rich semantic information extraction. For example, topic ontologies can be annotated within Protégé (Noy, Sintek, Decker, Crubezy, Fergerson, & Musen, 2001) WordNet tab. In addition, a query expansion algorithm based on ontologies needs to be developed for intelligent news information presentation.

CONCLUSION

Incremental text mining from news streams is an emerging technology as many news organizations are providing newswire services through the Internet. In order to accommodate dynamically changing topics, efficient incremental document clustering algorithms need to be developed. The algorithms must address the special requirements in news clustering, such as high rate of document update or ability to identify event level clusters, as well as topic level clusters.

In order to achieve rich semantic information retrieval within Web news services, an ontology-based approach would be provided. To overcome the problem of concept-based ontologies (i.e., topically related concepts and terms are not explicitly linked), topic ontologies are presented to characterize news topics at multiple levels of abstraction. In sum, coupling with topic ontologies and concept-based ontologies, support-

ing a topical search as well as semantic information retrieval can be achieved.

REFERENCES

Aggarwal, C.C., Gates, S.C., & Yu, P.S. (2004). On using partial supervision for text categorization. *IEEE Transactions on Knowledge and Data Engineering, 16*(2), 245-255.

Allan, J. (2002). Detection as multi-topic tracking. *Information Retrieval, 5*(2-3), 139-157.

Chung, S., & McLeod, D. (2003, November). Dynamic topic mining from news stream data. In *ODBASE'03* (pp. 653-670). Catania, Sicily, Italy.

Duda, R.O., Hart, P.E., & Stork D.G. (2001). *Pattern classification*. New York: Wiley Interscience.

Grosky, W.I., Sreenath, D.V., & Fotouhi, F. (2002). Emergent semantics and the multimedia semantic web. *SIGMOD Record, 31*(4), 54-58.

Hand, D.J., Mannila, H., & Smyth, P. (2001). *Principles of data mining (adaptive computation and machine learning)*. Cambridge, MA: The MIT Press.

Hatzivassiloglou, V., Gravano, L., & Maganti, A. (2000, July). An investigation of linguistic features and clustering algorithms for topical document clustering. In *ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'00)* (pp. 224-231). Athens, Greece.

Khan, L., McLeod, D., & Hovy, E. (2004). Retrieval effectiveness of an ontology-based model for information selection. *The VLDB Journal, 13*(1), 71-85.

Liu, X., Gong, Y., Xu, W., & Zhu, S. (2002, August). Document clustering with cluster refinement and model selection capabilities. In *ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR'02)* (pp. 91-198). Tampere, Finland.

Maedche, A., & Staab, S. (2001). Ontology learning for the semantic Web. *IEEE Intelligent Systems, 16*(2), 72-79.

Makkonen, J., Ahonen-Myka, H., & Salmenkivi, M. (2004). Simple semantics in topic detection and tracking. *Information Retrieval, 7*(3-4), 347-368.

Noy, N.F., Sintek, M., Decker, S., Crubezy, M., Fergerson, R.W., & Musen M.A. (2001). Creating semantic Web contents with Protégé 2000. *IEEE Intelligent Systems, 6*(12), 60-71.

Peng, F., & Schuurmans, D. (2003, April). Combining naive Bayes and n-gram language models for text classification. *European Conference on IR Research (ECIR'03)* (pp. 335-350). Pisa, Italy.

Tseng, Y. (2002). Automatic thesaurus generation for Chinese documents. *Journal of the American Society for Information Science and Technology, 53*(13), 1130-1138.

Zhao, Y., & Karypis, G. (2002, November). Evaluations of hierarchical clustering algorithms for document datasets. In *ACM International Conference on Information and Knowledge Management (CIKM'02)* (pp. 515-524). McLean, VA.

KEY TERMS

Clustering: An unsupervised process of dividing data into meaningful groups such that each identified cluster can explain the characteristics of underlying data distribution. Examples include characterization of different customer groups based on the customer's purchasing patterns, categorization of documents in the World Wide Web, or grouping of spatial locations of the earth where neighbor points in each region have similar short-term/long-term climate patterns.

Dynamic Topic Mining: A framework that supports the identification of meaningful patterns (e.g., events, topics, and topical relations) from news stream data.

First Story Detection: A TDT component that identifies whether a new document belongs to an existing topic or a new topic.

Ontology: A collection of concepts and inter-relationships.

Text Mining: A process of identifying patterns or trends in natural language text including document clustering, document classification, ontology learning, and etcetera.

Topic Detection And Tracking (TDT): Topic Detection and Tracking (TDT) is a DARPA-sponsored initiative to investigate the state of the art for news un-

derstanding systems. Specifically, TDT is composed of the following three major components: (1) segmenting a news stream (e.g., including transcribed speech) into topically cohesive stories; (2) identifying novel stories that are the first to discuss a new event; and (3) tracking known events given sample stories.

Topic Ontology: A collection of terms that characterize a topic at multiple levels of abstraction.

Topic Tracking: A TDT component that tracks events of interest based on sample news stories. It associates incoming news stories with the related stories, which were already discussed before or it monitors the news stream for further stories on the same topic.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 523-528, copyright 2005 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

Inexact Field Learning Approach for Data Mining

Honghua Dai
Deakin University, Australia

INTRODUCTION

Inexact fielding learning (IFL) (Ciesieski & Dai, 1994; Dai & Ciesieski, 1994a, 1994b, 1995, 2004; Dai & Li, 2001) is a rough-set, theory-based (Pawlak, 1982) machine learning approach that derives inexact rules from fields of each attribute. In contrast to a point-learning algorithm (Quinlan, 1986, 1993), which derives rules by examining individual values of each attribute, a field learning approach (Dai, 1996) derives rules by examining the fields of each attribute. In contrast to exact rule, an inexact rule is a rule with uncertainty. The advantage of the IFL method is the capability to discover high-quality rules from low-quality data, its property of low-quality data tolerant (Dai & Ciesieski, 1994a, 2004), high efficiency in discovery, and high accuracy of the discovered rules.

BACKGROUND

Achieving high prediction accuracy rates is crucial for all learning algorithms, particularly in real applications. In the area of machine learning, a well-recognized problem is that the derived rules can fit the training data very well, but they fail to achieve a high accuracy rate on new unseen cases. This is particularly true when the learning is performed on low-quality databases. Such a problem is referred as the Low Prediction Accuracy (LPA) problem (Dai & Ciesieski, 1994b, 2004; Dai & Li, 2001), which could be caused by several factors. In particular, overfitting low-quality data and being misled by them seem to be the significant problems that can hamper a learning algorithm from achieving high accuracy. Traditional learning methods derive rules by examining individual values of instances (Quinlan, 1986, 1993). To generate classification rules, these methods always try to find cut-off points, such as in well-known decision tree algorithms (Quinlan, 1986, 1993).

What we present here is an approach to derive rough classification rules from large low-quality numerical databases that appear to be able to overcome these two problems. The algorithm works on the fields of continuous numeric variables; that is, the intervals of possible values of each attribute in the training set, rather than on individual point values. The discovered rule is in a form called $\beta-$rule and is somewhat analogous to a decision tree found by an induction algorithm. The algorithm is linear in both the number of attributes and the number of instances (Dai & Ciesieski, 1994a, 2004).

The advantage of this inexact field-learning approach is its capability of inducing high-quality classification rules from low-quality data and its high efficiency that makes it an ideal algorithm to discover reliable knowledge from large and very large low-quality databases suitable for data mining, which needs higher discovering capability.

INEXACT FIELD-LEARNING ALGORITHM

Detailed description and the applications of the algorithm can be found from the listed articles (Ciesieski & Dai, 1994a; Dai & Ciesieski, 1994a, 1994b, 1995, 2004; Dai, 1996; Dai & Li, 2001; Dai, 1996). The following is a description of the inexact field-learning algorithm, the Fish_net algorithm:

Input: The input of the algorithm is a training data set with *m* instances and *n* attributes as follows:

Instances	X_1	X_2	...	X_n	Classes
$Instance_1$	a_{11}	a_{12}	...	a_{1n}	γ_1
$Instance_2$	a_{21}	a_{22}	...	a_{2n}	γ_2
...	...	...	...	...	...
$Instance_m$	a_{m1}	a_{m2}	...	a_{mn}	γ_m

(1)

Learning Process:

- **Step 1:** Work Out Fields of each attribute $\{x_i \mid 1 \le i \le n\}$ with respect to each class.

$$h_j^{(k)} = [h_{j_l}^{(k)}, h_{j_u}^{(k)}] \quad (k = 1,2,...,s;\ j = 1,2,...n). \tag{2}$$

$$h_{j_u}^{(k)} = \max_{a_{ij} \in a_j^{(k)}} \{a_{ij} \mid i = 1,2,...,m\} \quad (k = 1,2,...,s;\ j = 1,2,...n) \tag{3}$$

$$h_{j_l}^{(k)} = \min_{a_{ij} \in a_j^{(k)}} \{a_{ij} \mid i = 1,2,...,m\} \quad (k = 1,2,...,s;\ j = 1,2,...n). \tag{4}$$

- **Step 2:** Construct Contribution Function based on the fields found in Step 1.

$$\mu_{c_k}(x_j) = \begin{cases} 0 & x_j \in \bigcup_{i \ne k}^{s} h_j^{(i)} - h_j^{(k)} \\ 1 & x_j \in h_j^{(k)} - \bigcup_{i \ne k}^{s} h_j^{(i)} \\ \dfrac{x_j - a}{b - a} & x_j \in h_j^{(k)} \bigcap (\bigcup_{i \ne k}^{s} h_j^{(i)}) \end{cases} \quad (k = 1,2,...,s) \tag{5}$$

The formula (5) is given on the assumption that $[a,b] = h_j^{(k)} \bigcap (\bigcup_{i \ne k}^{s} h_j^{(i)})$, and for any small number $\varepsilon > 0, b \pm \varepsilon \in h_j^{(k)}$ and $a + \varepsilon \notin \bigcup_{i \ne k}^{s} h_j^{(i)}$ or $a - \varepsilon \notin \bigcup_{i \ne k}^{s} h_j^{(i)}$. Otherwise, the formula (5) becomes,

$$\mu_{c_k}(x_j) = \begin{cases} 0 & x_j \in \bigcup_{i \ne k}^{s} h_j^{(i)} - h_j^{(k)} \\ 1 & x_j \in h_j^{(k)} - \bigcup_{i \ne k}^{s} h_j^{(i)} \\ \dfrac{x_j - b}{a - b} & x_j \in h_j^{(k)} \bigcap (\bigcup_{i \ne k}^{s} h_j^{(i)}) \end{cases} \quad (k = 1,2,...,s) \tag{6}$$

- **Step 3:** Work Out Contribution Fields by applying the constructed contribution functions to the training data set.
  - Calculate the contribution of each instance.

$$\alpha(I_i) = (\sum_{j=1}^{n} \mu(x_{ij}))/n \quad (i = 1,2,...,m) \tag{7}$$

  - Work out the contribution field for each class $h^+ = < h_l^+, h_u^+ >$.

$$h_u^{(+)} = \max_{\alpha(I_i), I_i \in +} \{\alpha(I_i) \mid i = 1,2,...,m\} \tag{8}$$

$$h_l^{(+)} = \min_{\alpha(I_i), I_i \in +} \{\alpha(I_i) \mid i = 1,2,...,m\} \tag{9}$$

Similarly we can find $h^- = < h_l^-, h_u^- >$

- **Step 4:** Construct Belief Function using the derived contribution fields.

$$B_r(C) = \begin{cases} -1 & \text{Contribution} \in \text{NegativeRegion} \\ 1 & \text{Contribution} \in \text{PositiveRegion} \\ \dfrac{c - a}{b - a} & \text{Contribution} \in \text{RoughRegion} \end{cases} \tag{10}$$

- **Step 5:** Decide Threshold. It could have 6 different cases to be considered. The simplest case is to take the threshold

$$\alpha = \text{midpoint of } h^+ \text{and } h^- \tag{11}$$

- **Step 6:** Form the Inexact Rule.

$$\text{If} \quad \bar{\alpha}(I) = \frac{1}{N} \sum_{i=1}^{N} \mu(x_i) > \alpha$$
$$\text{Then} \quad \gamma = 1(B_r(c)) \tag{12}$$

This algorithm was tested on three large real observational weather data sets containing both high-quality and low-quality data. The accuracy rates of the forecasts were 86.4%, 78%, and 76.8%. These are significantly better than the accuracy rates achieved by C4.5 (Quinlan, 1986, 1993), feed forward neural networks, discrimination analysis, K-nearest neighbor classifiers, and human weather forecasters. The fish-net algorithm exhibited significantly less overfitting than the other algorithms. The training times were shorter, in some cases by orders of magnitude (Dai & Ciesieski, 1994a, 2004; Dai 1996).

FUTURE TRENDS

The inexact field-learning approach has led to a successful algorithm in a domain where there is a high

level of noise. We believe that other algorithms based on fields also can be developed. The $\beta-$rules, produced by the current FISH-NET algorithm involve linear combinations of attributes. Non-linear rules may be even more accurate.

While extensive tests have been done on the fish-net algorithm with large meteorological databases, nothing in the algorithm is specific to meteorology. It is expected that the algorithm will perform equally well in other domains. In parallel to most existing exact machine-learning methods, the inexact field-learning approaches can be used for large or very large noisy data mining, particularly where the data quality is a major problem that may not be dealt with by other data-mining approaches. Various learning algorithms can be created, based on the fields derived from a given training data set. There are several new applications of inexact field learning, such as Zhuang and Dai (2004) for Web document clustering and some other inexact learning approaches (Ishibuchi et al., 2001; Kim et al., 2003). The major trends of this approach are in the following:

1. Heavy application for all sorts of data-mining tasks in various domains.
2. Developing new powerful discovery algorithms in conjunction with IFL and traditional learning approaches.
3. Extend current IFL approach to deal with high dimensional, non-linear, and continuous problems.

CONCLUSION

The inexact field-learning algorithm: Fish-net is developed for the purpose of learning rough classification/forecasting rules from large, low-quality numeric databases. It runs high efficiently and generates robust rules that do not overfit the training data nor result in low prediction accuracy.

The inexact field-learning algorithm, fish-net, is based on fields of the attributes rather than the individual point values. The experimental results indicate that:

1. The fish-net algorithm is linear both in the number of instances and in the number of attributes. Further, the CPU time grows much more slowly than the other algorithms we investigated.
2. The Fish-net algorithm achieved the best prediction accuracy tested on new unseen cases out of all the methods tested (i.e., C4.5, feed-forward neural network algorithms, a k-nearest neighbor method, the discrimination analysis algorithm, and human experts.
3. The fish-net algorithm successfully overcame the LPA problem on two large low-quality data sets examined. Both the absolute LPA error rate and the relative LPA error rate (Dai & Ciesieski, 1994b) of the fish-net were very low on these data sets. They were significantly lower than that of point-learning approach, such as C4.5, on all the data sets and lower than the feed-forward neural network. A reasonably low LPA error rate was achieved by the feed-forward neural network but with the high time cost of error back-propagation. The LPA error rate of the KNN method is comparable to fish-net. This was achieved after a very high-cost genetic algorithm search.
4. The FISH-NET algorithm obviously was not affected by low-quality data. It performed equally well on low-quality data and high-quality data.

REFERENCES

Ciesielski, V., & Dai, H. (1994a). FISHERMAN: A comprehensive discovery, learning and forecasting systems. *Proceedings of 2nd Singapore International Conference on Intelligent System*, Singapore.

Dai, H. (1994c). *Learning of forecasting rules from large noisy meteorological data* [doctoral thesis]. RMIT, Melbourne, Victoria, Australia.

Dai, H. (1996a). Field learning. *Proceedings of the 19th Australian Computer Science Conference*.

Dai, H. (1996b). Machine learning of weather forecasting rules from large meteorological data bases. *Advances in Atmospheric Science*, *13*(4), 471-488.

Dai, H. (1997). *A survey of machine learning* [technical report]. Monash University, Melbourne, Victoria, Australia.

Dai, H. & Ciesielski, V. (1994a). Learning of inexact rules by the FISH-NET algorithm from low quality data. *Proceedings of the 7th Australian Joint Conference on Artificial Intelligence*, Brisbane, Australia.

Dai, H. & Ciesielski, V. (1994b). The low prediction accuracy problem in learning. *Proceedings of Second Australian and New Zealand Conference On Intelligent Systems*, Armidale, NSW, Australia.

Dai, H., & Ciesielski, V. (1995). *Inexact field learning using the FISH-NET algorithm* [technical report]. Monash University, Melbourne, Victoria, Australia.

Dai, H., & Ciesielski, V. (2004). *Learning of fuzzy classification rules by inexact field learning approach* [technical report]. Deakin University, Melbourne, Australia.

Dai, H. & Li, G. (2001). Inexact field learning: An approach to induce high quality rules from low quality data. *Proceedings of the 2001 IEEE International Conference on Data Mining (ICDM-01)*, San Jose, California.

Ishibuchi, H., Yamamoto, T., & Nakashima, T. (2001). Fuzzy data mining: Effect of fuzzy discretization. *Proceedings of IEEE International Conference on Data Mining*, San Jose, California.

Kim, M., Ryu, J., Kim, S., & Lee, J. (2003). Optimization of fuzzy rules for classification using genetic algorithm. *Proceedings of the 7th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Seoul, Korea.

Pawlak, Z. (1982). Rough sets. *International Journal of Information and Computer Science*, *11*(5), 145-172.

Quinlan, R. (1986). Induction of decision trees. *Machine Learning*, *1*, 81-106.

Quinlan, R. (1993). *C4.5: Programs for machine learning*. San Mateo, CA: Morgan Kaufmann Publishers.

Zhuang, L. & Dai, H. (2004). Maximal frequent itemset approach for Web document clustering. *Proceedings of the 2004 International Conference on Computer and Information Technology (CIT'04)*, Wuhan, China.

KEY TERMS

β-Rule: A type of inexact rule that represent the uncertainty with contribution functions and belief functions.

Exact Learning: The learning approaches that are capable of inducing exact rules.

Exact Rules: Rules without uncertainty.

Field Learning: Derives rules by looking at the field of the values of each attribute in all the instances of the training data set.

Inexact Learning: The learning by which inexact rules are induced.

Inexact Rules: Rules with uncertainty.

Low-Quality Data: Data with lots of noise, missing values, redundant features, mistakes, and so forth.

LPA (Low Prediction Accuracy) Problem: The problem when derived rules can fit the training data very well but fail to achieve a high accuracy rate on new unseen cases.

Point Learning: Derives rules by looking at each individual point value of the attributes in every instance of the training data set.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 611-614, copyright 2005 by IGI Publishing, formerly known as Idea Group Publishing (an imprint of IGI Global).

Information Fusion for Scientific Literature Classification

Gary G. Yen
Oklahoma State University, USA

INTRODUCTION

Scientific literatures can be organized to serve as a roadmap for researchers by pointing where and when the scientific community has been and is heading to. They present historic and current state-of-the-art knowledge in the interesting areas of study. They also document valuable information including author lists, affiliated institutions, citation information, keywords, etc., which can be used to extract further information that will assist in analyzing their content and relationship with one another. However, their tremendously growing size and the increasing diversity of research fields have become a major concern, especially for organization, analysis, and exploration of such documents. This chapter proposes an automatic scientific literature classification method (ASLCM) that makes use of different information extracted from the literatures to organize and present them in a structured manner. In the proposed ASLCM, multiple similarity information is extracted from all available sources and fused to give an optimized and more meaningful classification through using a genetic algorithm. The final result is used to identify the different research disciplines within the collection, their emergence and termination, major collaborators, centers of excellence, their influence, and the flow of information among the multidisciplinary research areas.

BACKGROUND

In addition to the body content, which is sometimes hard to analyze using a computer, scientific literatures incorporate essential information such as title, abstract, author, references and keywords that can be exploited to assist in the analysis and organization of a large collection (Singh, Mittal & Ahmad, 2007; Guo, 2007). This kind of analysis and organization proves helpful while dealing with a large collection of articles with a goal of attaining efficient presentation, visualization, and exploration in order to search for hidden information and useful connections lying within the collection. It can also serve as a historic roadmap that can be used to sketch the flow of information during the past and as a tool for forecasting possible emerging technologies. The ASLCM proposed in this study makes use of the above-mentioned types of information, which are available in most scientific literatures, to achieve an efficient classification and presentation of a large collection.

Many digital libraries and search engines make use of title, author, keyword, or citation information for indexing and cataloging purposes. Word-hit-based cataloging and retrieval using such types of information tends to miss related literatures that does not have the specified phrase or keyword, thus requiring the user to try several different queries to obtain the desired search result. In this chapter, these different information sources are fused to give an optimized and all-rounded view of a particular literature collection so that related literatures can be grouped and identified easily.

Title, abstract, author, keyword, and reference list are among the most common elements that are documented in typical scientific literature, such as a journal article. These sources of information can be used to characterize or represent literature in a unique and meaningful way, while performing computation for different information retrievals including search, cataloguing or organization. However, most of the methods that have been developed (Lawrence, Giles & Bollacker, 1999; Morris & Yen, 2004; White & McCain, 1998; Berry, Dramac & Jessup, 1999) use only one of these while performing the different information retrieval tasks, such as search and classification, producing results that focus only on a particular aspect of the collection. For example, usage of reference or citation information leads to a good understanding of the flow of information within the literature collection. This is because most literatures provide a link to the original base knowledge they used

within their reference list. In a similar fashion, use of information extracted from the authors list can lead to a good understanding of various author collaboration groups within the community along with their areas of expertise. This concept can be extended analogously to different information types provided by scientific literatures.

The theory behind the proposed ASLCM can be summarized and stated as follows:

- Information extracted from a particular field (e.g., citation alone) about a scientific literature collection conveys limited aspect of the real scenario.
- Most of the information documented in scientific literature can be regarded useful in some aspect toward revealing valuable information about the collection.
- Different information extracted from available sources can be fused to infer a generalized and complete knowledge about the entire collection.
- There lies an optimal proportion in which each source of information can be used in order to best represent the collection.

The information extracted from the above mentioned types of sources can be represented in the form of a matrix by using the vector space model (Salton, 1989). This model represents a document as a multi-dimensional vector. A set of selected representative features, such as keywords or citations, serves as a dimension and their frequency of occurrence in each article of interest is taken as the magnitude of that particular dimension, as shown in Equation (1).

$$M = \begin{array}{c} \\ D_1 \\ D_2 \\ \dots \\ D_n \end{array} \begin{array}{c} \begin{array}{cccc} T_1 & T_2 & \dots & T_t \end{array} \\ \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1t} \\ a_{21} & a_{22} & \dots & a_{21} \\ \dots & & & \\ a_{n1} & a_{n2} & \dots & a_{nt} \end{bmatrix} \end{array}. \quad (1)$$

In the above matrix, hereafter referred to as *adjacency matrix*, every row corresponds to a particular document and every column to a selected feature. For example, if t terms are chosen as representative features for n documents, $a_{i,j}$ corresponds to the frequency of occurrence of term j in document i. This technique of document modeling can be used to summarize and represent a collection of scientific literatures in terms of several adjacency matrices in an efficient manner. These adjacency matrices are later transformed into similarity matrices that measure the inter-document similarities so that classification, search, or other information retrieval tasks can be performed efficiently. The procedure of similarity matrix computation can be carried out by using either cosine coefficient, inner product, or dice coefficient (Jain, Murty & Flynn, 1999).

Selection of the type of similarity computation method to be used depends on the nature of the feature and the desired purpose. Once the different available inter-document similarity matrices are calculated, the information contained in each matrix can be fused into one generalized similarity matrix that can be used to classify and give a composite picture of the entire collection.

MAIN FOCUS

This chapter presents an information fusion scheme at the *similarity matrix* level in order to incorporate as much information as possible about the literature collection that would help better classify and discover hidden and useful knowledge. Information fusion is the concept of combining information obtained from multiple sources such as databases, sensors, human collected data, etc. in order to obtain a more precise and complete knowledge of a specific subject under study. The idea of information fusion is widely used in different areas such as image recognition, sensor fusion, information retrieval, etc. (Luo, Yih & Su, 2002).

Similarity Information Gathering

The scope of this research is mainly focused on similarity information extracted from bibliographic citations, author information, and word content analysis.

Bibliographic Citation Similarity: Given a collection of n documents and m references, an $n \times m$ paper-reference representation matrix PR can be formed, where P stands for paper and R for references. Here usually m tends to be much larger than n because a paper commonly cites more than one reference and different papers have different reference lists. An element of the PR matrix, $PR(i, j)$, is set to one if reference j is cited in paper i. As a result, this matrix is normally a sparse matrix with most of its entities having a value of zero. Having this PR matrix, the *citation similarity*

matrix can be calculated using the dice coefficient as follows:

$$S_r(i, j) = \frac{2 \times C_r(i, j)}{N_r(i) + N_r(j)} \quad (2)$$

where $S_r(i, j)$ is the citation similarity between documents i and j, $N_r(i)$ is the number of total references in document i, and C_r is a reference co-occurrence matrix, which can be calculated as $C_r = PR \times PR^T$. The value of $C_r(i, j)$ indicates the total number of common references between documents i and j.

Author Similarity: In a similar fashion, the *author similarity matrix* can be computed as follows,

$$S_a(i, j) = \frac{2 \times C_a(i, j)}{N_a(i) + N_a(j)}, \quad (3)$$

where $S_a(i,j)$ is the author similarity between documents i and j, $N_a(i)$ denotes the number of total authors in document i, and C_a is an author co-occurrence matrix, which can be calculated as $C_a = PA \times PA^T$. PA refers to the paper-author matrix defined in the same way as the PR matrix.

Term Similarity: The other similarity matrix constructed for the collection of the articles is the *term similarity matrix*. The steps taken toward the construction of this matrix are as follows: First, each word in the abstract of every article was parsed and entered into a database excluding a list of user-specified stop-words that did not bear any particular meaning. A basic word processing was also performed on the parsed words to avoid different versions of the same word by removing common prefix and suffixes such as *re*, *ing*, *ous*, etc. After this, the top t most frequent terms were selected as representing features for the document collection of interest. This value of the threshold was set depending on the total number of words extracted and the size of the document collection. Next, an $n \times t$ paper-term information matrix PT that contained the frequency or number of occurrence of each term in each document was constructed. $PT(i, j) = b$ implies that paper i contains term j b number of times. Next, the same approach as the previous ones was taken to calculate the term similarity matrix of the entire document collection as follows:

$$S_t(i, j) = \frac{2 \times C_t(i, j)}{N_t(i) + N_t(j)}, \quad (4)$$

where $S_t(i, j)$ is the term similarity between documents i and j, $N_t(i)$ is the number of selected terms in document i, and C_t is a term co-occurrence matrix, which can be calculated as $C_t = PT \times PT^T$.

The method of information fusion proposed in this chapter brings together the various similarity matrices obtained from different sources into one by linear weighting on each matrix as shown in Equation (5).

$$S_f = \sum_{i=1}^{n} w_i S_i, \quad (5)$$

where w_i is the weighting coefficient assigned to similarity matrix S_i by a genetic search algorithm and

$$\sum_{i=1}^{n} w_i = 1.$$

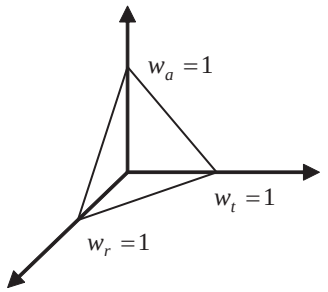
In the above case, $S_f = w_r \bullet S_r + w_a \bullet S_a + w_t \bullet S_t$ and $w_r + w_a + w_t = 1$.

Genetic Algorithms for Weighting Coefficient Search

The choice of the weighting coefficient assigned to each particular matrix is made depending on the content of information that it provides in order to attain *a good clustering*. However, there are infinite numbers of possible combinations from which the weighting parameters can be chosen, especially when dealing with a large number of similarity matrices. Figure 1 shows the input space of the weighting parameters for combining three similarity matrices (i.e., $n = 3$). Every point lying in the plane shown in this figure is a possible candidate for this particular problem, and this makes the search for the optimal weighting parameters even more complex.

This chapter proposes a genetic algorithm (GA) (Goldberg, 1989; Yen & Lu, 2003) based weighting parameter search method that can be effectively used to search for the best weighting parameters even in the case where there are a large number of similarity matrices to be fused. GA is a population-based, point-by-point search and optimization algorithm with

Figure 1. Input space of the weighting coefficients



a characteristic of avoiding local minima, which is the major challenge of optimization problems with a higher number of dimensions. GA encodes candidate solutions to a particular problem in the form of binary bits, which are later given the chance to undergo genetic operations such as reproduction, mutation, crossover, etc. At every generation a fitness value is calculated for each individual member, and those individuals with higher fitness values are given more chance to reproduce and express their schema until a stopping criterion is reached. This stopping criterion can be either a maximum number of generations or achievement of the minimum desired error for the criterion function that is being evaluated. Table 1 gives the general flow structure of a typical GA. The actual coding implementation of the genes for searching n weighting coefficients is shown in Figure 2.

In Figure 2, $B_{i,j}$ refers to the j^{th} binary bit of the i^{th} weighting parameter, which is calculated as:

$$w_i = \frac{(B_{i1}B_{i2}...B_{im})_2}{2^m}, \qquad (6)$$

where m is the number of bits used to encode the coefficients. The choice for the value of m is determined by the required resolution of the parameters, i.e., larger choice of m gives more variety of choices but with increasing level of computational complexity. In this coding, the n^{th} weighting coefficient is obtained by

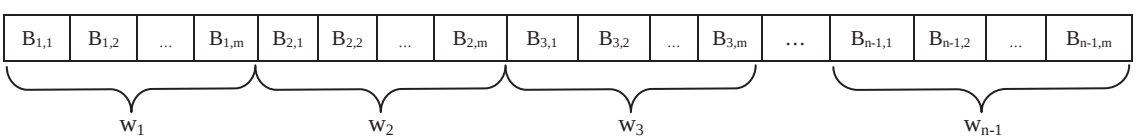
$$w_n = 1 - \sum_{i=1}^{n-1} w_i .$$

Table 1. Flow structure of a typical genetic algorithm

1. Set iteration index i to 0.
2. Generate $P(i)$ number of populations at random.
3. *REPEAT*
   a. Evaluate the fitness of each individual in $P(i)$.
   b. Select parents from $P(i)$ based on a fitness criterion function.
   c. Produce next generation $P(i+1)$ using genetic operations.
   d. Set $i=i+1$.

   UNTIL the stopping criterion is met.

Figure 2. Genetic encoding for n-1 similarity weighting coefficients



Performance Evaluation

Having the above genetic structure setup, the composite similarity matrix resulting from each individual member of the population is passed to an agglomerative hierarchical classification routine (Griffiths, Robinson & Willett, 1984), and its performance is evaluated using the technique described below until the GA reaches its stopping criteria. Classification, in a broad sense, is the act of arranging or categorizing a set of elements on the basis of a certain condition (Gordon, 1998). In this particular problem that we are dealing with, the criterion for classifying a collection of scientific literatures is mainly to be able to group documents that belong to related areas of research based solely upon features, such as citations, authorship and keywords. The method introduced here to measure the performance of literature classification is *minimization of the Discrete Pareto Distribution coefficient*. Pareto Distribution (Johnson, Kotz & Kemp, 1992) is a type of distribution with heavy tail and exists in many natural incidents. In this distribution function, large occurrences are very rare and few occurrences are common. Figure 3 shows an example of the citation pattern of documents obtained in the area of Micro-Electro Mechanical Systems (MEMS) (Morris *et al*., 2003). In this example collection, almost 18,000 papers were cited only once, and there are only few papers that received a citation-hit greater than five.

Plotting the above distribution on a log-log scale gives the result shown in Figure 4. From the graph we see that a linear curve can be fit through the data, and the slope of the curve is what is referred to as the Pareto-Distribution Coefficient (γ).

Measuring clustering performance in terms of minimization of the Pareto Distribution coefficient of citation trends directly relates to clustering documents that have common reference materials together. The slope of the line fit through the log-log plot decreases as more and more documents with common references are added to the same group. This criterion function can thus be used to measure a better classification of the literatures according to their research area.

The overall structure of this classification architecture is shown in Figure 5. As can be seen in the diagram, different similarity information matrices that are extracted from the original data are fused according to the parameters provided by the GA, and performance evaluation feedback is carried on until a stopping criterion is met. Upon termination of the GA, the best result is taken for visualization and interpretation.

Presentation

A hierarchical time line (Morris *et al*., 2003; Morris, Asnake & Yen, 2003) map visualization technique is used to present the final result of the classification, and exploration and interpretation are performed on it. In this hierarchical time line mapping technique (as shown in Figure 6), documents are plotted as circles on a date versus cluster membership axis. Their size is made to correspond to the relative number of times

Figure 3. Frequency (f) versus number of papers cited f times

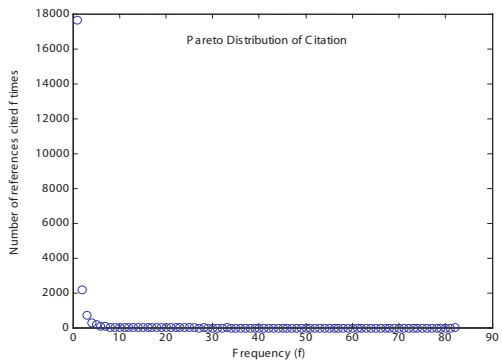


Figure 4. Log-log plot of frequency (f) versus number of papers cited f times

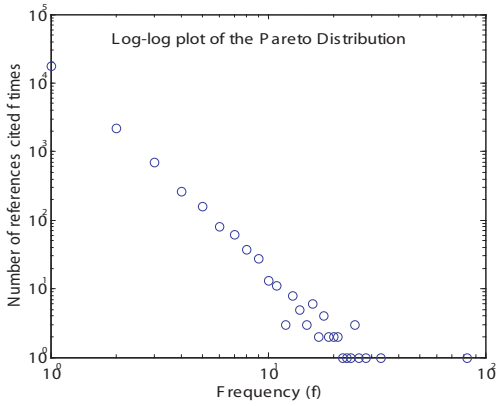
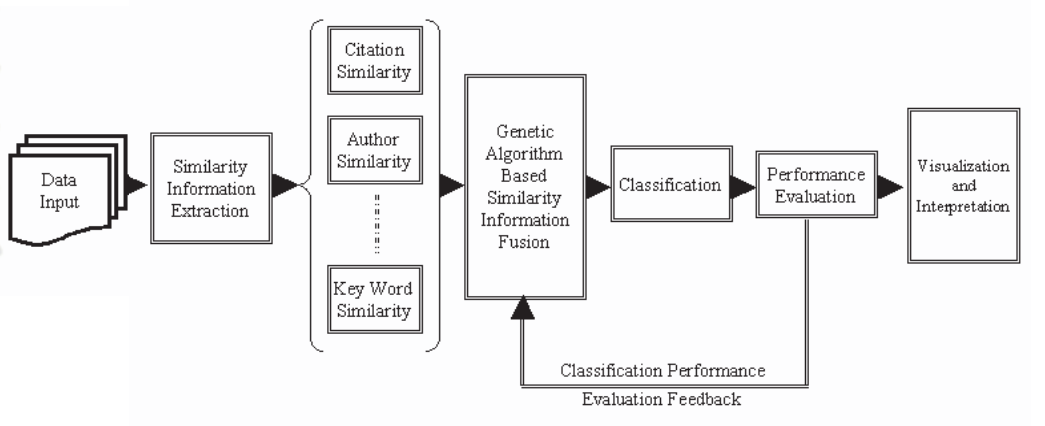


Figure 5. Proposed classification architecture

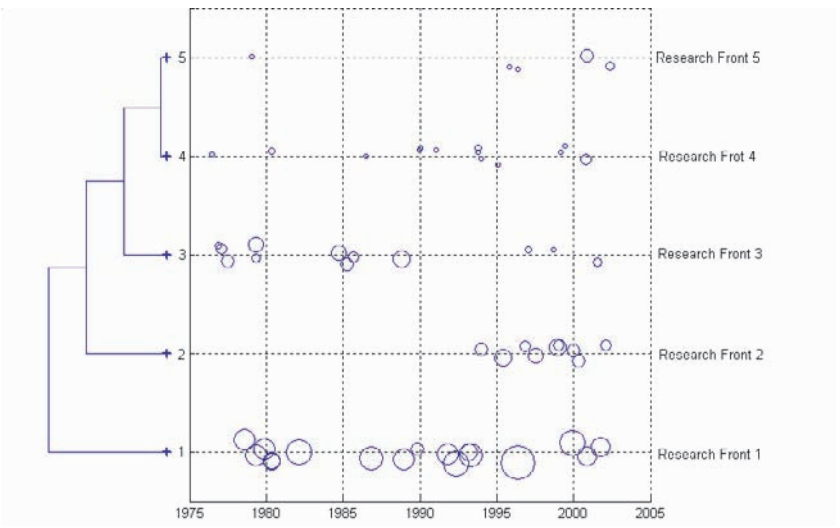


each was cited. Hence, large circles indicate documents that have been heavily cited and vice versa. Each horizontal line corresponds to a cluster that refers to a particular research area. The tree structure on the left side of the map shows the hierarchical organization and relationship between the clusters formed. The labels to the right of the map are produced manually by taking a closer note at the titles and most frequent words of the documents belonging to that particular cluster. This representation also provides temporal information that can be used to investigate the historic timing of the different research areas.

CASE STUDY

This section presents a case study performed on the subject of anthrax using the information fusion technique of scientific literature classification proposed in this chapter. The study was performed based on a collection of articles obtained from the ISI Science Citation Index library using the query phrase "anthrax anthracis." This query returned 2,472 articles published from early 1945 to the beginning of 2003. The summary of this collection is given in Table 2.

Figure 6. Hierarchical time line layout



Standard procedures were applied to filter and store these articles into a Microsoft Access database for later use. Out of the returned 2,472 articles only those articles that had five or more citation links to others were considered for further analysis. As a result, the number of articles under the study was reduced to 987. This helped exclude documents that were not strongly connected to others. Three similarity matrices out of the citation, author, and keyword frequency information as discussed above were constructed and passed to the genetic algorithm based optimization routine, which resulted in 0.78, 0.15, and 0.07 weighting coefficients for w_r, w_a, and w_t, respectively. The population size is chosen to be 50, each with total number of bits equal to 30. The stopping criterion based on Pareto distribution coefficient is set for $\gamma = 2.0$, which is chosen heuristically based upon a few trials and errors. This is in a similar spirit as one would choose a maximum number of generations to evolve the genetic algorithm. The ultimate classification result is presented in a visualization map, which is then subjected to personal preference. This result was used to classify the 987 documents, and the map shown in Figure 7 was obtained.

This map was then manually labeled after studying the title, abstract, and keyword content of the documents belonging to each cluster. One and two word frequency tables were also generated to cross check the labels generated with the most frequent words within each cluster. The labeled version of the map is shown in Figure 8. The result shown in Figure 8 identifies the different research areas within the collection as shown in the labels. Also, the important documents are shown as larger circles, of which two by Friedlander and Leppla are labeled. Friedlander's paper related to macrophages is cited 173 times within this collection and Leppla's finding on the edema factor, one part of the anthrax toxin, was cited 188 times. These two papers are among the few seminal papers in this collection where significant findings were documented. It can also be noted that Friedlander's finding opened a new dimension in the area of protective antigen study that started after his publication in 1986.

Table 2. Summary of the collected articles on anthrax

Total number of	Count
Papers	2,472
References	25,007
Authors	4,493

The similarity connection between documents was also analyzed in order to identify the most connected clusters and the flow of information among the different research areas. This is shown in Figure 9. Thin lines connect those documents that have a citation similarity value greater than a threshold value of 0.3. From this connection we can draw a conclusion about the flow of information among the different research areas shown by the dashed line. This line starts from the cluster on *preliminary anthrax research* extending until the latest development on biological warfare, which includes all the publications on bioterrorism related to the 2001 US postal attacks.

We can also see the active period of the research areas. For example, most research on preliminary anthrax was done from mid 1940s to early 1970s. Interestingly, research on bioterrorism is shown to be active from late 1990s to present.

The effect of Leppla's finding was also further studied and is shown in Figure 10. The dotted lines connect Leppla's article to its references and the solid lines connect it to those articles citing Leppla's paper. This figure shows that Leppla's finding was significant and has been used by many of the other identified research areas.

The following example shows a summary of the collaboration of Leppla, who has 74 publications in this collection. Table 5.2 below gives a summary of those people with whom Leppla published at least seven times.

FUTURE TRENDS

Future research on the technique proposed to classify scientific literatures includes application of the concept of similarity information fusion to a collection of free information sources such as news feeds, Web pages, e-mail messages, etc., in order to identify the hidden trends and links lying within. This requires the incorporation of natural language processing techniques to extract word frequency information for similarity matrix computation. In addition, significant effort should be made to automate the complete process, from literature collection, to classification and visual-

Figure 7. A first look at the classification result

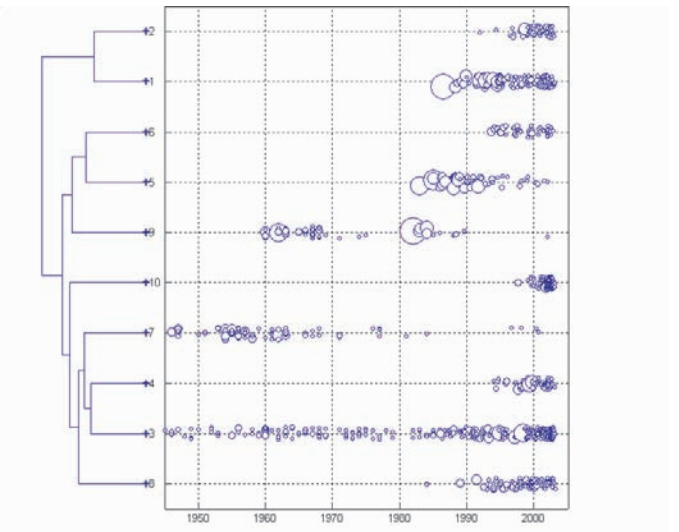


Figure 8. Labelled map

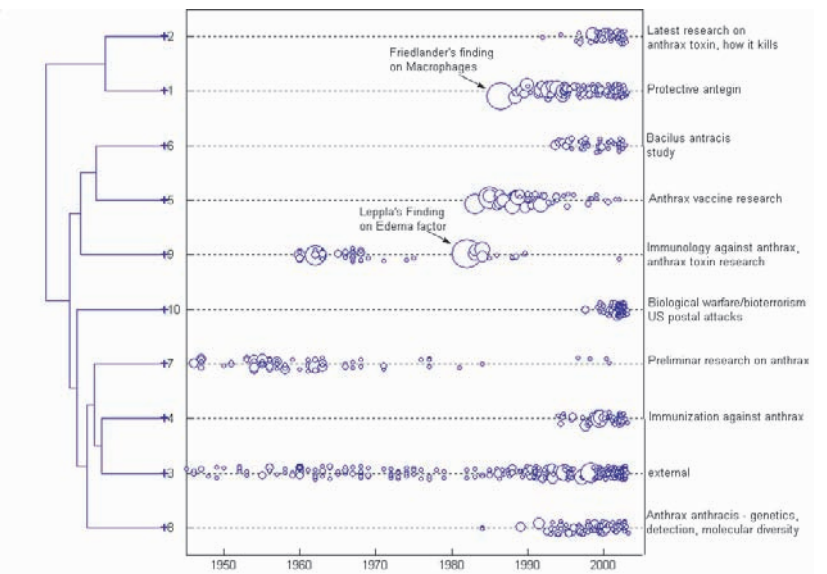


Figure 9. Flow of information among the different research areas

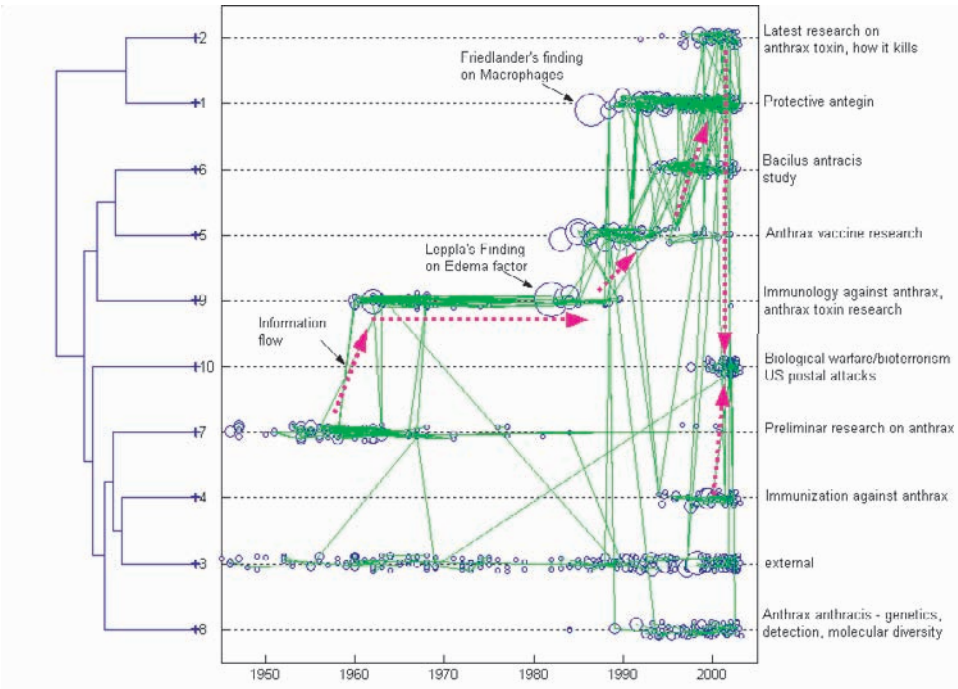


Figure 10. Influence of Leppla's finding

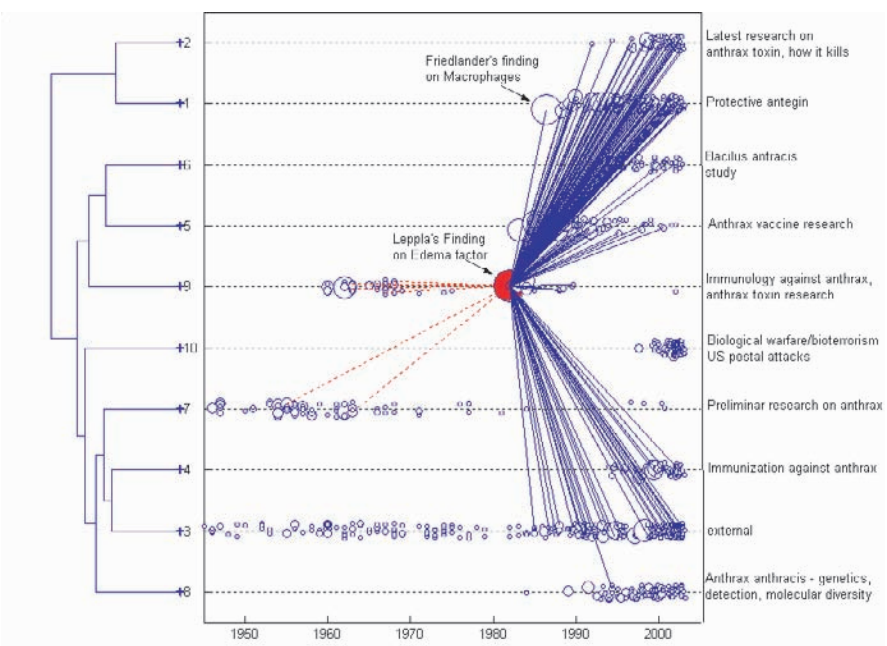


Table 3. Major collaborators of LEPPLA

Author	Count
KLIMPEL, KR	16
SINGH, Y	12
ARORA, N	9
Liu, SH	8
LITTLE, SF	8
FRIEDLANDER, AM	7
GORDON, VM	7

ization, to interpretation and reasoning. Additionally, computational complexity analysis of the proposed algorithm is necessary to justify its applications to real-world problems.

CONCLUSION

The method presented in this chapter aimed at the identification of the different research areas, their time of emergence and termination, major contributing authors, and the flow of information among the different research areas by making use of different available information within a collection of scientific literatures. Similarity information matrices were fused to give a generalized composite similarity matrix based on optimal weighting parameters obtained by using a genetic algorithm based search method. Minimization of the Pareto distribution coefficient was used as a measure for achieving a good clustering. A good cluster, in the case of scientific classification, contains many documents that share the same set of reference materials or base knowledge. The final classification result was presented in the form of a hierarchical time line, and further exploration was carried out on it. From the result, it was possible to clearly identify the different areas of research, major authors and their radius of influence, and the flow of information among the different research areas. The results clearly show the dates of emergence and termination of specific research disciplines.

REFERENCES

Berry, M. W., Dramac, Z., & Jessup, E. R. (1999). Matrices, vector spaces, and information retrieval, *Society for Industrial and Applied Mathematics*, 41, 335-362.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. New York, NY: Addison-Wesley.

Gordon, A. D. (1998). *Classification*. Baton Rouge, LA: Campman & Hall/CRC.

Griffiths, A., Robinson, L. A., & Willett, P. (1984). Hierarchic agglomerative clustering methods for automatic document classification, *Journal of Documentation*, 40, 175-205.

Guo, G. (2007). A computer-aided bibliometric system to article ranked lists in interdisciplinar generate core subjects. Information Sciences, 177, 3539-3556.

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: a review, *ACM Computing Surveys*, 31, 264-322.

Johnson, N. L., Kotz, S., & Kemp, A. W. (1992). *Univariate Discrete Distributions*. New York, NY: John Wiley & Sons.

Lawrence, S., Giles, C. L., & Bollacker, K. (1999). Digital libraries and autonomous citation indexing, *IEEE Computer*, 32, 67-71.

Luo, R. C., Yih, C. C., & Su, K. L. (2002). Multisensor fusion and integration: approaches, applications, and future research directions, *IEEE Sensors Journal*, 2, 107-119.

Morris, S. A., Asnake, B., & Yen, G. G. (2003). Dendrogram seriation using simulated annealing, *International Journal of Information Visualization*, 2, 95-104.

Morris, S. A., Yen, G. G., Wu, Z., & Asnake, B. (2003). Timeline visualization of research fronts, *Journal of the American Society for Information Science and Technology*, 54, 413-422.

Morris, S. A., & Yen, G. G. (2004). Crossmaps: visualization of overlapping relationships in collections of journal papers, *Proceedings of National Academy of Science*, 101(14), 5291-5296.

Salton, G. (1989). *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. New York, NY: Addison-Wesley.

Singh, G., Mittal, R., & Ahmad, M. A. (2007). A bliometric study of literature on digital libraries. Electronic Library, 25, 342-348.

White, H. D., & McCain, K. W. (1998). Visualizing a discipline: an author co-citation analysis of information science, 1972-1995, *Journal of the American Society for Information Science and Technology*, 49, 327-355.

Yen, G. G. & Lu, H. (2003). Rank and density-based genetic algorithm for multi-criterion optimization, *IEEE Transactions on Evolutionary Computations*, 7, 325-343.

KEY TERMS

Bibliometrics: Bibliometrics is a set of methods used to study or measure texts and information. Citation analysis and content analysis are commonly used bibliometric methods. While bibliometric methods are most often used in the field of library and information science, bibliometrics have wide applications in other areas. In fact, many research fields use bibliometric methods to explore the impact of their field, the impact of a set of researchers, or the impact of a particular paper.

Genetic Algorithm: A search technique used in computing to find true or approximate solutions to optimization and search problems. Genetic algorithms are categorized as global search heuristics. Genetic algorithms are a particular class of evolutionary algorithms that use techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover (also called recombination).

Information Fusion: Refers to the field of study of techniques attempting to merge information from disparate sources despite differing conceptual, contextual and typographical representations. This is used in data mining and consolidation of data from semi- or unstructured resources.

Information Visualization: Information visualization is a branch of computer graphics and user interface design that are concerned with presenting data to users, by means of interactive or animated digital images. The goal of this area is usually to improve understanding of the data being presented.

Pareto Distribution: The Pareto distribution, named after the Italian economist Vilfredo Pareto, is a power law probability distribution that coincides with social, scientific, geophysical, and many other types of observable phenomena.

Statistical Classification: Statistical classification is a statistical procedure in which individual items are placed into groups based on quantitative information on one or more characteristics inherent in the items (referred to as traits, variables, characters, etc) and based on a training set of previously labeled items.

Information Veins and Resampling with Rough Set Theory

Benjamin Griffiths
Cardiff University, UK

Malcolm J. Beynon
Cardiff University, UK,

INTRODUCTION

Rough Set Theory (RST), since its introduction in Pawlak (1982), continues to develop as an effective tool in data mining. Within a set theoretical structure, its remit is closely concerned with the classification of objects to decision attribute values, based on their description by a number of condition attributes. With regards to RST, this classification is through the construction of '*if .. then ..*' decision rules. The development of RST has been in many directions, amongst the earliest was with the allowance for miss-classification in the constructed decision rules, namely the Variable Precision Rough Sets model (VPRS) (Ziarko, 1993), the recent references for this include; Beynon (2001), Mi et al. (2004), and Ślęzak and Ziarko (2005). Further developments of RST have included; its operation within a fuzzy environment (Greco et al., 2006), and using a dominance relation based approach (Greco et al., 2004).

The regular major international conferences of 'International Conference on Rough Sets and Current Trends in Computing' (RSCTC, 2004) and 'International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing' (RSFDGrC, 2005) continue to include RST research covering the varying directions of its development. This is true also for the associated book series entitled 'Transactions on Rough Sets' (Peters and Skowron, 2005), which further includes doctoral theses on this subject. What is true, is that RST is still evolving, with the eclectic attitude to its development meaning that the definitive concomitant RST data mining techniques are still to be realised. Grzymala-Busse and Ziarko (2000), in a defence of RST, discussed a number of points relevant to data mining, and also made comparisons between RST and other techniques.

Within the area of data mining and the desire to identify relationships between condition attributes, the effectiveness of RST is particularly pertinent due to the inherent intent within RST type methodologies for data reduction and feature selection (Jensen and Shen, 2005). That is, subsets of condition attributes identified that perform the same role as all the condition attributes in a considered data set (termed β-reducts in VPRS, see later). Chen (2001) addresses this, when discussing the original RST, they state it follows a reductionist approach and is lenient to inconsistent data (contradicting condition attributes - one aspect of underlying uncertainty). This encyclopaedia article describes and demonstrates the practical application of a RST type methodology in data mining, namely VPRS, using nascent software initially described in Griffiths and Beynon (2005). The use of VPRS, through its relative simplistic structure, outlines many of the rudiments of RST based methodologies.

The software utilised is oriented towards 'hands on' data mining, with graphs presented that clearly elucidate 'veins' of possible information identified from β-reducts, over different allowed levels of miss-classification associated with the constructed decision rules (Beynon and Griffiths, 2004). Further findings are briefly reported when undertaking VPRS in a resampling environment, with leave-one-out and bootstrapping approaches adopted (Wisnowski et al., 2003). The importance of these results is in the identification of the more influential condition attributes, pertinent to accruing the most effective data mining results.

BACKGROUND

VPRS development on RST is briefly described here (see Ziarko, 1993; Beynon, 2001). It offers the allowance for miss-classification of objects in the constructed decision rules (determined by a β value over its allowed domain, see later), as well as correct classification and

the non-classification of objects (the β value infers a level of certainty in the model). This is one of a number of directions of development in RST based research. By way of example, the Bayesian rough set model moves away from the requirement for a particular β value (Ślęzak and Ziarko, 2005), instead it considers an appropriate certainty gain expression (using Bayesian reasoning). The relative simplicity of VPRS offers the reader the opportunity to perceive the appropriateness of RST based approaches to data mining.

Central to VPRS (and RST) is the information system (termed here as the data set), which contains a universe of objects U (o_1, o_2, …), each characterised by a set condition attributes C (c_1, c_2, …) and classified to a set of decision attributes D (d_1, d_2, …). Through the indiscernibility of objects based on C and D, respective condition and decision classes of objects are found. For a defined proportionality value β, the β-positive region corresponds to the union of the set of condition classes (using a subset of condition attributes P), with conditional probabilities of allocation to a set of objects Z (using a decision class $Z \in E(D)$), which are at least equal to β. More formally:

β-positive region of the set $Z \subseteq U$ and $P \subseteq C$: $POS_P^{\beta}(Z) = \bigcup_{\Pr(Z \mid X_i) \geq \beta} \{X_i \in E(P)\}$,

where β is defined here to lie between 0.5 and 1 (Beynon, 2001), and contributes to the context of a *majority inclusion* relation. That is, those condition classes in a β-positive region, $POS_P^{\beta}(Z)$, each have a majority of objects associated with the decision class $Z \in E(D)$. The numbers of objects included in the condition classes that are contained in the respective β-positive regions for each of the decision classes, subject to the defined β value, make up a measure of the *quality of classification*, denoted $\gamma^{\beta}(P, D)$, and given by:

$$\gamma^{\beta}(P, D) = \frac{\text{card}(\bigcup_{Z \in E(D)} POS_P^{\beta}(Z))}{\text{card}(U)},$$

where $P \subseteq C$. The $\gamma^{\beta}(P, D)$ measure with the β value means that for the objects in a data set, a VPRS analysis may define them in one of three states; not classified, correctly classified and miss-classified. Associated with this is the $\beta_{\min}$ value, the lowest of the (largest) proportion values of β that allowed the set of condition classes to be in the β-positive regions constructed. That is, a β value above this upper bound would imply at least one of the contained condition classes would then not be given a classification.

VPRS further applies these defined terms by seeking subsets of condition attributes (termed β-*reducts*), capable of explaining the associations given by the whole set of condition attributes, subject to the majority inclusion relation (using a β value). Within data mining, the notion of a β-reduct is directly associated with the study of data reduction and feature selection (Jensen and Shen, 2005). Ziarko (1993) states that a β-reduct (R) of the set of conditional attributes C, with respect to a set of decision attributes D, is:

i) A subset R of C that offers the same quality of classification, subject to the β value, as the whole set of condition attributes.

ii) No proper subset of R has the same quality of the classification as R, subject to the associated β value.

An identified β-reduct is then used to construct the decision rules, following the approach utilised in Beynon (2001). In summary, for the subset of attribute values that define the condition classes associated with a decision class, the values that discern them from the others are identified. These are called prime implicants and form the condition parts of the constructed decision rules (further reduction in the prime implicants also possible).

Main Focus

When large data sets are considered, the identification of β-reducts and adopted balance between classification/miss-classification of objects infers small veins of relevant information are available within the whole β domain of (0.5, 1]. This is definitive of data mining and is demonstrated here using the VPRS software introduced in Griffiths and Beynon (2005). A large European bank data set is utilised to demonstrate the characteristics associated with RST in data mining (using VPRS).

The Bank Financial Strength Rating (BFSR) introduced by Moody's rating agency is considered (Moody's, 2004), which represent their opinion of a bank's intrinsic safety and soundness (see Poon et al., 1999). Thirteen BFSR levels exist, but here a more general grouping of ratings of (A or B), C and (D or E) are considered (internally labelled 0 to 2). This article considers an exhaustive set of 309 European banks for which a BFSR rating has been assigned to them. The numbers of banks assigned a specific rating is; (A or B)

- 102, C - 162 and (D or E) - 45, which indicates a level of imbalance in the allocation of ratings to the banks. This issue of an unbalanced data set has been considered within RST, see Grzymala-Busse et al. (2003).

Nine associated financial variables were selected from within the bank data set, with limited concern for their independence or association with the BFSR rating of the European banks, denoted by; C1 - Equity/Total Assets, C2 - Equity/Net Loans, C3 - Equity/Dep. & St. Funding, C4 - Equity/Liabilities, C5 - Return on average assets, C6 - Return on average equity, C7 - Cap. Funds/Liabilities, C8 - Net Loans/ Total Assets and C9 - Net Loans/ Cust. & St. Funding. With VPRS a manageable granularity of the considered data set affects the number and specificity of the resultant decision rules constructed (Grzymala-Busse and Ziarko, 2000). Here a simple equal-frequency intervalisation is employed, which intervalises each condition attribute into two groups, in this case with evaluated cut-points; C1 (5.53), C2 (9.35), C3 (8.18), C4 (5.96), C5 (0.54), C6 (10.14), C7 (8.4), C8 (60.25) and C9 (83.67).

Using the subsequently produced discretised data set, a VPRS analysis, would firstly consider the identification of β-reducts (see Figure 1).

The presented software snapshot in Figure 1 shows veins of solid lines in each row, along the β domain, associated with subsets of condition attributes and where they would be defined as β-reducts. With respect to data mining, an analyst is able to choose any such vein, from which a group of decision rules are constructed that describes the accrued information (see later), because by definition they offer the same information as the whole data set. The decision rules associated with the different 'solid line' sub-domains offer varying insights into the information inherent in the considered data set.

The VPRS resampling based results presented next, offer further information to the analyst(s), in terms of importance of the condition attributes, and descriptive statistics relating to the results of the resampling process. Firstly, there is a need to have a criterion to identify a single β-reduct in each resampling run. The criterion considered here (not the only approach), considers those β-reducts offering the largest possible level of quality of classification (those associated with the solid lines towards the left side of the graph in Figure 1), then those which have the higher β_{min} values (as previously defined), then those β-reducts based on the least number of condition attributes included in the β-reducts and finaly those based on the largest associated β sub-domains.

The first resampling approach considered is 'leave-one-out' (Weiss & Kulikowski, 1991), where within each run, the constructed classifier (set of decision rules in VPRS) is based on $n - 1$ objects (the training set), and tested on the remaining single object (the test set). There are subsequently n runs in this process, for every run the classifier is constructed on nearly all the objects, with each object at some point during the process used as a test object. For sample sizes of over 100, leave-one-out is considered an accurate, almost completely unbiased, estimator of the true error rate (Weiss & Kulikowski, 1991). In the case of the BFSR bank data set, 309 runs were undertaken and varying results available, see Figure 2.

In Figure 2, the top table exposits descriptive statistics covering relevant measures in the 'leave-one-out'

Figure 1. Example VPRS analysis 'information veins' on BFSR bank data set

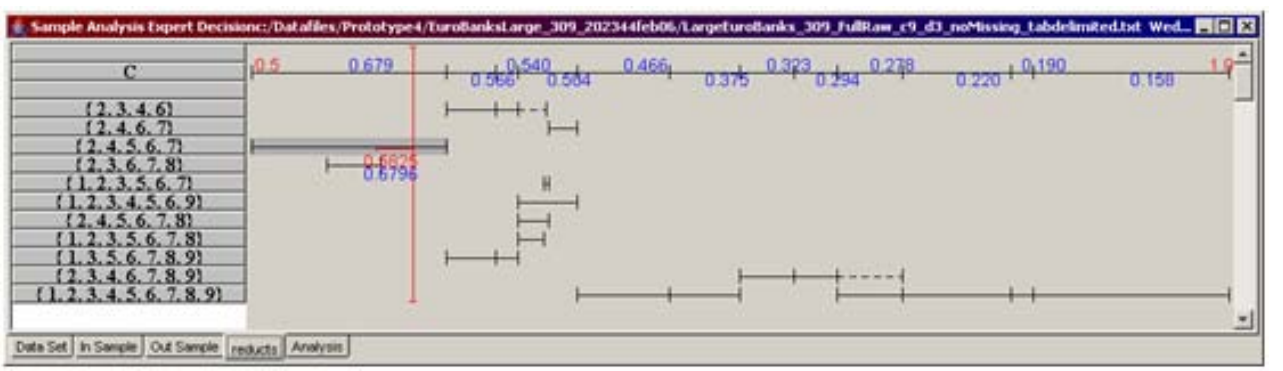
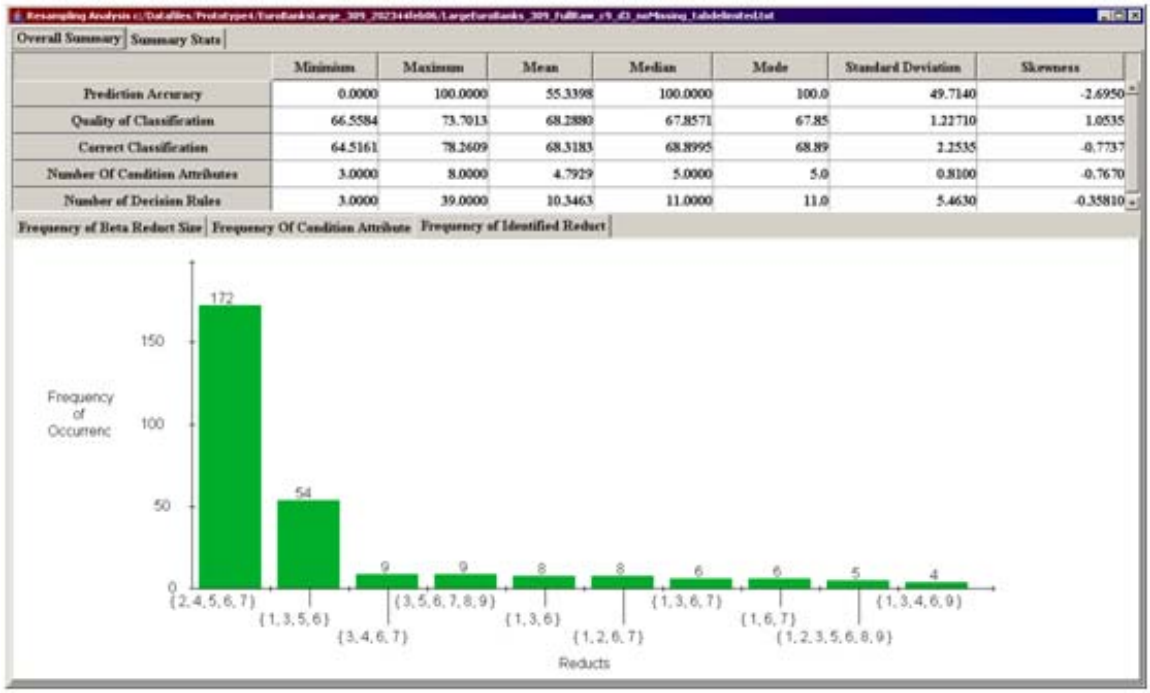


Figure 2. Descriptive statistics and one histogram from 'leave-one-out' analysis



VPRS analysis. The first measure is predictive accuracy, next and pertinent to VPRS is the quality of classification (number of objects assigned a classification) and then below this is a measure of the number of objects that are asigned a correct classification. The remaining two rows are significant to the individual β-reducts identified in the runs performed, namely the average number of condition attributes included and average number of associated decision rules constructed.

The presented histogram lists the top ten most frequently identified β-reducts, the most frequent being {C2, C4, C5, C6, C7}, identified in 172 out of the 309 runs performed. This β-reduct includes five condition attributes, near to the mean number of 4.7929 found from the 309 runs in the whole 'leave-one-out' analysis.

The second resampling approach considered is 'bootstrapping' (Wisnowski et al., 2003), which draws a random sample of *n* objects, from a data set of size *n*, using sampling with replacement. This constitutes the training set, objects that do not appear within the training set constitute the test set. On average the proportion of the objects appearing in the original data set and the training set is 0.632 (0.368 are therefore duplicates of them), hence the average proportion of the testing set is 0.368. One method that yields strong results is the 0.632B bootstrap estimator (Weiss & Kulikowski, 1991). In the case of the BFSR bank data set, 500 runs were undertaken and varying results available, see Figure 3.

The descriptive statistics reported in the top table in Figure 3 are similar to those found from the leave-one-out analysis, but the presented histogram showing the top ten most frequently identified β-reducts is noticeably different from the previous analysis. This highlights that the training data sets constructed from the two resampling approaches must be inherently different with respect to advocating the importance of different subsets of condition attributes (β-reducts).

Following the 'leave-one-out' findings, the β-reduct {C2, C4, C5, C6, C7} was further considered on the whole data set, importantly it is present in the graph presented in Figure 1, from which a set of decision rules was constructed, see Figure 4.

The presented decision rules in Figure 4 are a direct consequence from the data mining undertaken on all possible solid line veins of pertinent information elucidated in Figure 1 (using the evidence from the leave-one-out analysis). To interpret the decision rules presented, rule 2 is further written in full.

Figure 3. Descriptive statistics and one histogram from 'bootstrapping' analysis with 500 runs

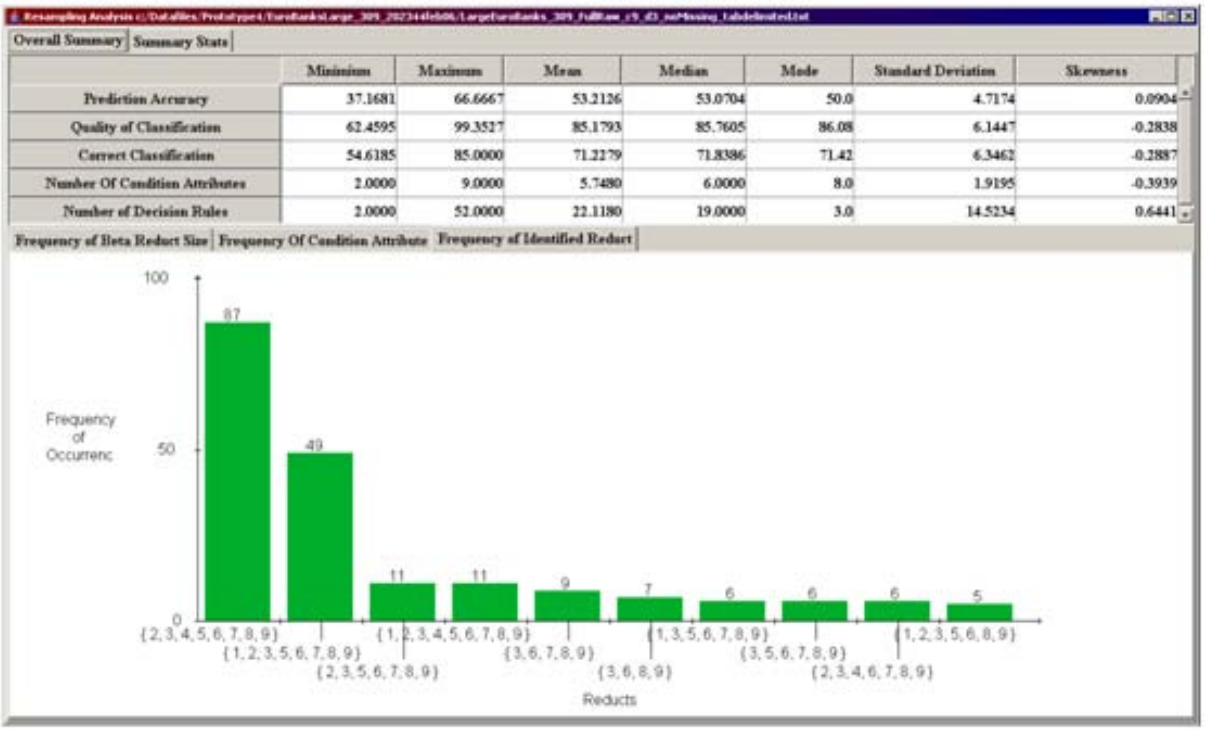
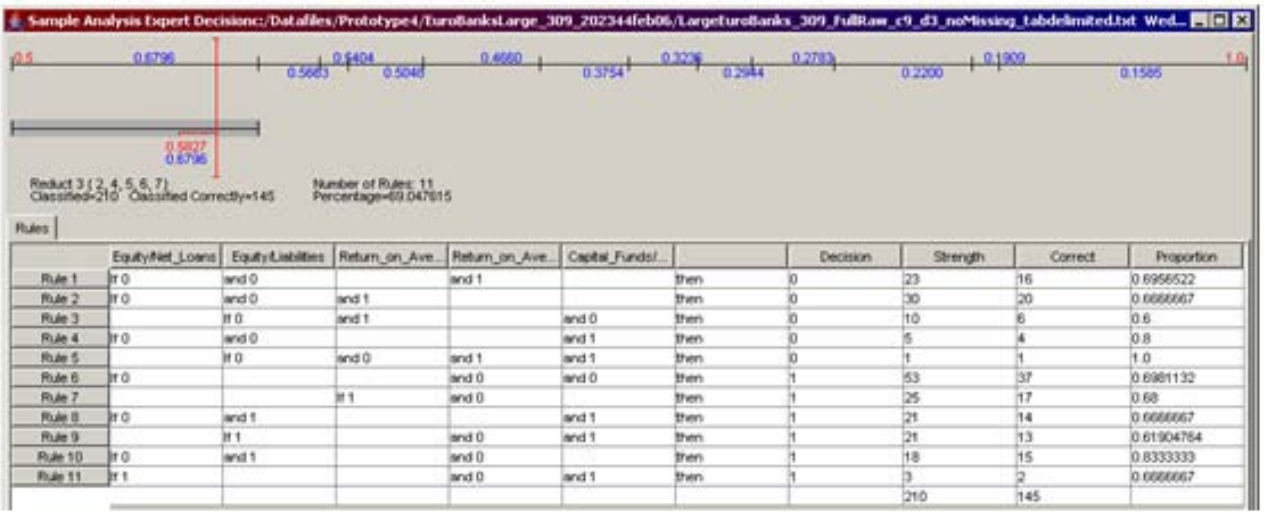


Figure 4. Decision rules associated with β-reduct {C2, C4, C5, C6, C7}



"If 'Equity/Net Loans' is less than 9.35, 'Equity/Liabilities' is less than 8.18 and 'Return on average equity' greater than or equal to 10.14 then BFSR rating is (A or B), which classifies 30 banks of which 20 were correctly classified (having a 66.66% accuracy)."

This level of interpretability available from a VPRS analysis is most important to effective data mining.

FUTURE TRENDS

The development of data mining techniques needs to keep in touch with the requirements and desires of the relevant analyst(s). Further, they need to offer analyst(s) every opportunity to inspect the data set, including any varying directions of investigation available. The

VPRS approach, while the more open of the RST variations to follow, through the prescribed software certainly allows for these requirements. Perhaps the role of Occam's razor may play more importance in future developments, with the simplicity of techniques allowing the analyst(s) to understand why the results are what they are.

CONCLUSION

Data mining, while often a general term for the analysis of data, is specific in its desire to identify relationships within the attributes present in a data set. Techniques to undertake this type of investigation have to offer practical information to the analyst(s). Rough Set Theory (RST) certainly offers a noticeable level of interpretability to any findings accrued in terms of the decision rules constructed, this is true for any of its nascent developments also. In the case of Variable Precision Rough Sets Model (VPRS), the allowed miss-classification in the decision rules may purport more generality to the interpretations given.

In the desire for feature selection, the notion of reducts (β-reducts in VPRS) fulfils this role effectively, since their identification is separate to the construction of the associated decision rules. That is, their identification is more concerned with obtaining similar levels of quality of classification of the data set with the full set of condition attributes. Within a resampling environment VPRS (and RST in general), offers a large amount of further information to the analyst(s). This includes the importance of the individual condition attributes, and general complexity of the data set, in terms of number of condition attributes in the identified β-reducts and number of concomitant decision rules constructed.

REFERENCES

Beynon, M. (2001). Reducts within the variable precision rough set model: A further investigation. *European Journal of Operational Research*, 134, 592-605.

Beynon, M. J., & Griffiths, B. (2004). An Expert System for the Utilisation of the Variable Precision Rough Sets Model. *The Fourth International Conference on Rough Sets and Current Trends in Computing*, Lecture Notes in Artificial Intelligence, LNAI 3066, Springer Verlag, 714-720.

Chen, Z. (2001). *Data mining and uncertain reasoning: An integrated approach*. John Wiley, New York.

Greco, S., Matarazzo, B., & Słowiński, R. (2004). Axiomatic characterization of a general utility function and its particular cases in terms of conjoint measurement and rough-set decision rules. *European Journal of Operational Research, 158*(2), 271-292.

Greco, S., Inuiguchi, M., & Słowiński, R. (2006). Fuzzy rough sets and multiple-premise gradual decision rules. *International Journal of Approximate Reasoning, 41*(2), 179-211.

Griffiths, B., & Beynon, M.J. (2005). Expositing stages of VPRS analysis in an expert system: Application with bank credit ratings. *Expert Systems with Applications, 29*(4), 879-888.

Grzymala-Busse, J. W., & Ziarko, W. (2000). Data mining and rough set theory. *Communications of the ACM, 43*(4), 108-109.

Grzymala-Busse, J. W., Goodwin, L. K., Grzymala-Busse, W. J., & Zheng, X. (2003). An approach to imbalanced data sets based on changing rule strength. In S. K. Pal, L. Polkowski & A. Skowron (Eds.), *Rough-Neural Computing* (pp. 543-553), Springer Verlag.

Jensen, R., & Shen, Q. (2005). Fuzzy-rough data reduction with ant colony optimization. *Fuzzy Sets and Systems*, 149, 5-20.

Mi, J.-S., Wu, W.-Z., & Zhang, W.-X. (2004). Approaches to knowledge reduction based on variable precision rough set model, *Information Sciences*, 159, 255-272.

Moody's. (2004) Rating Definitions - Bank Financial Strength Ratings, Internet site www.moodys.com. Accessed on 23/11/2004.

Pawlak, Z. (1982). Rough sets. *International Journal of Information and Computer Sciences, 11*(5), 341-356.

Peters, J. F., & Skowron, A. (2005) Transactions on Rough Sets II. Springer, Heidelberg, Germany.

Poon, W. P. H., Firth, M., & Fung, H.-G. (1999). A multivariate analysis of the determinants of Moody's bank financial strength ratings. *Journal of International Financial Markets*, 9, 267-283.

RSCTC (2004) *The 4th International Conference on Rough Sets and Current Trends in Computing*, Lecture Notes in Artificial Intelligence, LNAI 3066, Springer Verlag, ISBN 3-540-221117-4.

RSFDGrC (2005). *The 10th International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing (Part I)*, Lecture Notes in Artificial Intelligence, LNAI 3641, Springer Verlag, ISBN 3-540-28653-5.

Ślęzak, D., & Ziarko, W. (2005). The Investigation of the Bayesian Rough Set Model, *International Journal of Approximate Reasoning*, 40, 81-91.

Weiss, S. M., & Kulikowski, C. A. (1991). *Computer Systems that Learn: Classification and prediction methods from statistics, neural nets, machine learning, and expert systems*. California, Morgan Kaufmann.

Wisnowski, J. W., Simpson, J. R., Montgomery, D.C., & Runger, G.C. (2003). Resampling methods for variable selection in robust regression. *Computational Statistics & Data Analysis*, 43, 341-355.

Ziarko, W. (1993). Variable precision rough set model. *Journal of Computer and System Sciences*, 46, 39-59.

KEY TERMS

Bootstrapping: Resampling approach whereby a random sample of *n* objects are drawn from a data set of size *n* (using sampling with replacement), which constitutes the training set, objects that do not appear within the training set constitute the test set.

Decision Rules: Set of decision rules of the form 'if … then …'.

Feature Selection: Subsets of condition attributes identified and used to perform the same role as all the condition attributes in a considered data set.

Indiscernibility Relation: Process of grouping objects based on having the same series of attributes values.

Leave-One-Out: Resampling approach whereby each single object is left out as the testing set and the classifiers built on the remaining objects that make up the associated training set.

Miss-Classification: The incorrect classification of an object to one decision class, when known to be associated with another.

Quality of Classification: Number of objects assigned a classification (correctly or incorrectly).

Reduct: Subset of condition attributes that offers the same quality of classification as the whole set.

VPRS: Variable Precision Rough Sets Model.

Instance Selection

Huan Liu
Arizona State University, USA

Lei Yu
Arizona State University, USA

INTRODUCTION

The amounts of data become increasingly large in recent years as the capacity of digital data storage worldwide has significantly increased. As the size of data grows, the demand for data reduction increases for effective data mining. Instance selection is one of the effective means to data reduction. This article introduces basic concepts of instance selection, its context, necessity and functionality. It briefly reviews the state-of-the-art methods for instance selection.

Selection is a necessity in the world surrounding us. It stems from the sheer fact of limited resources. No exception for data mining. Many factors give rise to data selection: data is not purely collected for data mining or for one particular application; there are missing data, redundant data, and errors during collection and storage; and data can be too overwhelming to handle. Instance selection is one effective approach to data selection. It is a process of choosing a subset of data to achieve the original purpose of a data mining application. The ideal outcome of instance selection is a model independent, minimum sample of data that can accomplish tasks with little or no performance deterioration.

BACKGROUND AND MOTIVATION

When we are able to gather as much data as we wish, a natural question is "how do we efficiently use it to our advantage?" Raw data is rarely of direct use and manual analysis simply cannot keep pace with the fast accumulation of massive data. Knowledge discovery and data mining (KDD), an emerging field comprising disciplines such as databases, statistics, machine learning, comes to the rescue. KDD aims to turn raw data into nuggets and create special edges in this ever competitive world for science discovery and business intelligence. The KDD process is defined (Fayyad *et al.*, 1996) as *the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data*. It includes data selection, preprocessing, data mining, interpretation and evaluation. The first two processes (data selection and preprocessing) play a pivotal role in successful data mining (Han and Kamber, 2001). Facing the mounting challenges of enormous amounts of data, much of the current research concerns itself with scaling up data mining algorithms (Provost and Kolluri, 1999). Researchers have also worked on scaling down the data - an alternative to the algorithm scaling-up. The major issue of scaling down data is to select the relevant data and then present it to a data mining algorithm. This line of work is in parallel with the work on algorithm scaling-up and the combination of the two is a two-edged sword in mining nuggets from massive data.

In data mining, data is stored in a *flat file* and described by terms called *attributes* or *features*. Each line in the file consists of attribute-values and forms an *instance*, also named as a *record*, *tuple*, or *data point* in a multi-dimensional space defined by the attributes. Data reduction can be achieved in many ways (Liu and Motoda, 1998; Blum and Langley, 1997; Liu and Motoda, 2001). By selecting features, we reduce the number of columns in a data set; by discretizing feature-values, we reduce the number of possible values of features; and by selecting instances, we reduce the number of rows in a data set. We focus on instance selection here.

Instance selection reduces data and enables a data mining algorithm to function and work effectively with huge data. The data can include almost everything related to a domain (recall that data is not solely collected for data mining), but one application is normally about using one aspect of the domain. It is natural and sensible to focus on the relevant part of the data for the application so that search is more focused and mining is more efficient. It is often required to clean data before mining. By selecting relevant instances, we can

usually remove irrelevant, noise, and redundant data. The high quality data will lead to high quality results and reduced costs for data mining.

MAJOR LINES OF RESEARCH AND DEVELOPMENT

A spontaneous response to the challenge of instance selection is, without fail, some form of sampling. Although it is an important part of instance selection, there are other approaches that do not rely on sampling, but resort to search or take advantage of data mining algorithms. In the following, we start with sampling methods, and proceed to other instance selection methods associated with data mining tasks such as classification and clustering.

Sampling Methods

Sampling methods are useful tools for instance selection (Gu, Hu, and Liu, 2001).

$\binom{N}{n}$ *Simple random sampling* is a method of selecting n instances out of the N such that every one of the distinct samples has an equal chance of being drawn. If an instance that has been drawn is removed from the data set for all subsequent draws, the method is called random sampling without replacement. Random sampling with replacement is entirely feasible: at any draw, all N instances of the data set are given an equal chance of being drawn, no matter how often they have already been drawn.

Stratified random sampling The data set of N instances is first divided into subsets of N_1, N_2,…, N_l instances, respectively. These subsets are non-overlapping, and together they comprise the whole data set (i.e., $N_1+N_2,\ldots,+N_l=N$). The subsets are called strata. When the strata have been determined, a sample is drawn from each stratum, the drawings being made independently in different strata. If a simple random sample is taken in each stratum, the whole procedure is described as stratified random sampling. It is often used in applications that we wish to divide a heterogeneous data set into subsets, each of which is internally homogeneous.

Adaptive sampling refers to a sampling procedure that selects instances depending on results obtained from the sample. The primary purpose of adaptive sampling is to take advantage of data characteristics in order to obtain more precise estimates. It takes advantage of the result of preliminary mining for more effective sampling and vise versa.

Selective sampling is another way of exploiting data characteristics to obtain more precise estimates in sampling. All instances are first divided into partitions according to some homogeneity criterion, and then random sampling is performed to select instances from each partition. Since instances in each partition are more similar to each other than instances in other partitions, the resulting sample is more representative than a randomly generated one. Recent methods can be found in (Liu, Motoda, and Yu, 2002) in which samples selected from partitions based on data variance result in better performance than samples selected from random sampling.

Methods for Labeled Data

One key data mining application is classification – predicting the class of an unseen instance. The data for this type of application is usually labeled with class values. Instance selection in the context of classification has been attempted by researchers according to the classifiers being built. We include below five types of selected instances.

Critical points are the points that matter the most to a classifier. The issue was originated from the learning method of Nearest Neighbor (NN) (Cover and Thomas, 1991). NN usually does not learn during the training phase. Only when it is required to classify a new sample does NN search the data to find the nearest neighbor for the new sample and use the class label of the nearest neighbor to predict the class label of the new sample. During this phase, NN could be very slow if the data is large and be extremely sensitive to noise. Therefore, many suggestions have been made to keep only the critical points so that noisy ones are removed as well as the data set is reduced. Examples can be found in (Yu *et al.*, 2001) and (Zeng, Xing, and Zhou, 2003) in which critical data points are selected to improve the performance of collaborative filtering.

Boundary points are the instances that lie on borders between classes. Support vector machines (SVM) provide a principled way of finding these points through minimizing structural risk (Burges, 1998). Using a non-linear function φ to map data points to a high-dimensional feature space, a non-linearly separable

data set becomes linearly separable. Data points on the boundaries, which maximize the margin band, are the support vectors. Support vectors are instances in the original data sets, and contain all the information a given classifier needs for constructing the decision function. Boundary points and critical points are different in the ways how they are found.

Prototypes are representatives of groups of instances via averaging (Chang, 1974). A prototype that represents the typicality of a class is used in characterizing a class, instead of describing the differences between classes. Therefore, they are different from critical points or boundary points.

Tree based sampling Decision trees (Quinlan, 1993) are a commonly used classification tool in data mining and machine learning. Instance selection can be done via the decision tree built. In (Breiman and Friedman, 1984), they propose *delegate sampling*. The basic idea is to construct a decision tree such that instances at the leaves of the tree are approximately uniformly distributed. Delegate sampling then samples instances from the leaves in inverse proportion to the density at the leaf and assigns weights to the sampled points that are proportional to the leaf density.

Instance labeling In real world applications, although large amounts of data are potentially available, the majority of data are not labeled. Manually labeling the data is a labor intensive and costly process. Researchers investigate whether experts can be asked to only label a small portion of the data that is most relevant to the task if it is too expensive and time consuming to label all data. Usually an expert can be engaged to label a small portion of the selected data at various stages. So we wish to select as little data as possible at each stage, and use an adaptive algorithm to guess what else should be selected for labeling in the next stage. Instance labeling is closely associated with adaptive sampling, clustering, and active learning.

Methods for Unlabeled Data

When data is unlabeled, methods for labeled data cannot be directly applied to instance selection. The widespread use of computers results in huge amounts of data stored without labels (web pages, transaction data, newspaper articles, email messages) (Baeza-Yates and Ribeiro-Neto, 1999). Clustering is one approach to finding regularities from unlabeled data. We discuss three types of selected instances here.

Prototypes are pseudo data points generated from the formed clusters. The idea is that after the clusters are formed, one may just keep the prototypes of the clusters and discard the rest data points. The *k*-means clustering algorithm is a good example of this sort. Given a data set and a constant *k*, the *k*-means clustering algorithm is to partition the data into *k* subsets such that instances in each subset are similar under some measure. The *k* means are iteratively updated until a stopping criterion is satisfied. The prototypes in this case are the *k* means.

Prototypes plus sufficient statistics In (Bradley, Fayyad, and Reina, 1998), they extend the *k*-means algorithm to perform clustering in one scan of the data. By keeping some points that defy compression plus some sufficient statistics, they demonstrate a scalable *k*-means algorithm. From the viewpoint of instance selection, it is a method of representing a cluster using both defiant points and pseudo points that can be reconstructed from sufficient statistics, instead of keeping only the *k* means.

Squashed data are some pseudo data points generated from the original data. In this aspect, they are similar to prototypes as both may or may not be in the original data set. Squashed data points are different from prototypes in that each pseudo data point has a weight and the sum of the weights is equal to the number of instances in the original data set. Presently two ways of obtaining squashed data are (1) model free (DuMouchel *et al*., 1999) and (2) model dependent (or likelihood based (Madigan *et al*., 2002)).

FUTURE TRENDS

As shown above, instance selection has been studied and employed in various tasks (sampling, classification, and clustering). Each task is very unique in itself as each task has different information available and requirements. It is clear that a universal model of instance selection is out of the question. This short article provides some starting points that can hopefully lead to *more concerted study and development of new methods for instance selection*. Instance selection deals with scaling down data. When we understand better instance selection, it is natural to investigate if *this work can be combined with other lines of research* in overcoming the problem of huge amounts of data, such as algorithm scaling-up, feature selection and construction. It is a big challenge

to integrate these different techniques in achieving the common goal - effective and efficient data mining.

CONCLUSION

With the constraints imposed by computer memory and mining algorithms, we experience selection pressures more than ever. The central point of instance selection is *approximation*. Our task is to achieve as good mining results as possible by approximating the whole data with the selected instances and hope to do better in data mining with instance selection as it is possible to remove noisy and irrelevant data in the process. In this short article, we have presented an initial attempt to review and categorize the methods of instance selection in terms of sampling, classification, and clustering.

REFERENCES

Baeza-Yates, R. & Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. Addison Wesley and ACM Press.

Blum, A. & Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97, 245–271.

Bradley, P., Fayyad, U., & Reina, C. (1998). Scaling clustering algorithms to large databases. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, pp. 9 – 15.

Burges, C. (1998). A tutorial on support vector machines. *Journal of Data Mining and Knowledge Discovery*, 2,121-167.

Chang, C. (1974). Finding prototypes for nearest neighbor classifiers. *IEEE Transactions on Computers*, C-23.

Cover, T. M. & Thomas, J. A. (1991). *Elements of Information Theory*. Wiley.

DuMouchel, W., Volinsky, C., Johnson, T., Cortes, C., & Pregibon, D. (1999). Squashing flat files flatter. In *Proceedings of the Fifth ACM Conference on Knowledge Discovery and Data Mining*, pp. 6-15.

Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., & Uthurusamy, R. (1996). From data mining to knowledge discovery. In *Advances in Knowledge Discovery and Data Mining*.

Gu, B., Hu, F., & Liu, H. (2001). Sampling: knowing whole from its part. In *Instance Selection and Construction for Data Mining*. Boston: Kluwer Academic Publishers.

Han, J. & Kamber, M. (2001). *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers.

Liu, H. & Motoda, H., (1998). *Feature Selection for Knowledge Discovery and Data Mining*. Boston: Kluwer Academic Publishers.

Liu, H. & Motoda, H., editors (2001). *Instance Selection and Construction for Data Mining*. Boston: Kluwer Academic Publishers.

Liu, H., Motoda, H., & Yu, L. (2002). Feature selection with selective sampling. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pp. 395-402, 2002.

Madigan, D., Raghavan, N., DuMouchel, W., Nason, M., Posse, C., & Ridgeway, G. (2002). Likelihood-based data squashing: a modeling approach to instance construction. *Journal of Data Mining and Knowledge Discovery*, 6(2), 173-190.

Provost, F. & Kolluri, V. (1999). A survey of methods for scaling up inductive algorithms. *Journal of Data Mining and Knowledge Discovery*, 3, 131 – 169.

Quinlan, R.J. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann.

Yu, K., Xu, X., Ester, M., & Kriegel, H. (2001) Selecting relevant instances for efficient and accurate collaborative filtering. In *Proceedings of the Tenth International Conference on Information and Knowledge Management*, pp. 239-46.

Zeng, C., Xing, C., & Zhou, L. (2003). Similarity measure and instance selection for collaborative filtering. In *Proceedings of the Twelfth International Conference on World Wide Web*, pp. 652-658.

KEY TERMS

Classification: A process of predicting the classes of unseen instances based on patterns learned from available instances with predefined classes.

Clustering: A process of grouping instances into clusters so that instances are similar to one another within a cluster but dissimilar to instances in other clusters.

Data Mining: The application of analytical methods and tools to data for the purpose of discovering patterns, statistical or predictive models, and relationships among massive data.

Data Reduction: A process of removing irrelevant information from data by reducing the number of features, instances, or values of the data.

Instance: A vector of attribute-values in a multi-dimensional space defined by the attributes, also named as a record, tuple, or data point.

Instance Selection: A process of choosing a subset of data to achieve the original purpose of a data mining application as if the whole data is used.

Sampling: A procedure that draws a sample S_i by a random process in which each S_i receives its appropriate probability P_i of being selected.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 621-624, copyright 2005 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

Integration of Data Mining and Operations Research

Stephan Meisel
University of Braunschweig, Germany

Dirk C. Mattfeld
University of Braunschweig, Germany

INTRODUCTION

Basically, Data Mining (DM) and Operations Research (OR) are two paradigms independent of each other. OR aims at optimal solutions of decision problems with respect to a given goal. DM is concerned with secondary analysis of large amounts of data (Hand et al., 2001). However, there are some commonalities. Both paradigms are application focused (Wu et al., 2003; White, 1991). Many Data Mining approaches are within traditional OR domains like logistics, manufacturing, health care or finance. Further, both DM and OR are multidisciplinary. Since its origins, OR has been relying on fields such as mathematics, statistics, economics and computer science. In DM, most of the current textbooks show a strong bias towards one of its founding disciplines, like database management, machine learning or statistics.

Being multidisciplinary and application focused, it seems to be a natural step for both paradigms to gain synergies from integration. Thus, recently an increasing number of publications of successful approaches at the intersection of DM and OR can be observed. On the one hand, efficiency of the DM process is increased by use of advanced optimization models and methods originating from OR. On the other hand, effectiveness of decision making is increased by augmentation of traditional OR approaches with DM results. Meisel and Mattfeld (in press) provide a detailed discussion of the synergies of DM and OR.

BACKGROUND

The aim of DM is identification of models or patterns representing relationships in data. The DM process was shaped by two milestones. First, the discovery of relationships in data was established as a multi-step procedure (Fayyad et al., 1996). Secondly, a framework for the core DM step, including the definition of DM tasks was specified (Hand et al., 2001). One of the tasks is exploratory data analysis by means of interactive and visual techniques. Further, descriptive modeling aims at describing all of the data. Predictive modeling comprises classification and regression methods. The task of discovering patterns and rules focuses on particular aspects of the data instead of giving a description of the full data set at hand. Finally retrieval by content addresses the search for similar patterns in text and image datasets.

According to a DM task, a model or pattern structure is selected. The DM algorithm determines an instance of the structure best fitting a given set of target data. The target data set is tailored to the requirements of the DM algorithm in a preprocessing step modifying an initial set of collected data.

OR approaches require identification of decision variables and definition of a pursued objective. A decision model is developed, specifying the set of feasible values of the decision variables and an objective function. A search procedure is then applied in order to determine the optimal values for the decision variables. In case the structure of the decision model does not allow for efficient search methods, the decision model structure is often replaced by heuristic decision rules allowing for deliberate selection of a solution.

The use of optimization methods originating from OR is established since long for at least some of the DM tasks. Mangasarian (1997) discusses the relevance of mathematical programming for large-scale DM problems. A summary of early works in the field is given by Bradley et al. (1999). An application domain

specific article by Padmanabhan and Tuzhilin (2003) gives an overview on the use of optimization for DM in electronic customer relationship management.

However, the multitude of new developments at the intersection of DM and OR does not only comprise more advanced optimization models for DM. Rather, many bare improved OR approaches by integration of DM.

MAIN FOCUS

Regarding recent advances published in literature three types of synergies of DM and OR can be distinguished. On the one hand, application of optimization methods to increase DM efficiency. On the other hand, the use of DM to increase OR effectiveness either by improvement of a decision model structure or by improvement of decision model. Each of the three synergies is discussed below.

Increased Efficiency

Optimization problems may be encountered at several points in both of the major DM steps. Some of the works from literature focus on preprocessing operations. However, most papers are about the use of OR for efficient implementation of descriptive and predictive modeling, explorative data analysis as well as the discovery of patterns and rules.

1. *Preprocessing*—Preprocessing is split into a series of problems of different complexity. Some of these may not be met seriously without the application of OR methods. Examples are the feature subset selection problem, the discretization of a continuous domain of attribute values and de-duplication of information in databases.

Yang and Olafsson (2005) formulate the feature subset selection problem as combinatorial optimization problem. For solution they apply the nested partitions metaheuristic. Pendharkar (2006) considers feature subset selection as a constraint satisfaction optimization problem and proposes a hybrid heuristic based on simulated annealing and artificial neural networks. Meiri and Zahavi (2006) also apply simulated annealing to combinatorial feature subset selection and outperform the traditional stepwise regression method.

Janssens et al. (2006) model a discretization problem as shortest path network and solve it by integer programming.

OR-based de-duplication is considered by Spiliopoulos and Sofianopoulou (2007). They model the problem of calculating dissimilarity between data records as modified shortest path problem offering new insights into the structure of de-duplication problems.

2. *Descriptive Modeling*—The most common technique for the DM task of descriptive modeling is cluster analysis. Boginsiki et al. (2006) formulate the clustering problem as NP-hard clique partitioning problem and give a heuristic allowing for efficient solution.

Saglam et al. (2006) develop a heuristic for solving a mixed integer model for clustering. They show the procedure to outperform the well known k-means algorithm in terms of accuracy. Beliakov and King (2006) formulate the fuzzy c-means algorithm as a bi-level optimization problem and solve it by a discrete gradient method. The algorithm is capable of identifying non-convex overlapped d-dimensional clusters, a property present in only a few experimental methods before.

Kulkarni and Fathi (in press) apply a branch and cut algorithm to an integer programming model for clustering and find the quality of its LP relaxation to depend on the strength of the natural clusters present. Hence, they specify the conditions for an optimal solution to be expected by a branch and cut algorithm. Innis (2006) builds an integer program for seasonal clustering taking into account the time order of data.

3. *Predictive Modeling*—A number of recent publications offer elaborate approaches for the task of predictive modeling by use of OR-methods. Üney and Türkay (2006) present a multi-class data classification method based on mixed-integer programming. Exceeding traditional methods, they introduce the concept of hyperboxes for defining class boundaries increasing both classification accuracy and efficiency.

Jones et al. (2007) present a goal programming model allowing for flexible handling of the two class classification problem. The approach pursues both, maximization of the level of correct classifications and minimization of the level of misclassifications.

Carrizosa and Martin-Barragan (2006) address the same problem. They formulate a biobjective optimization problem and derive the set of pareto optimal solutions as well as a mechanism to select one out of these solutions. Trafalis and Gilbert (2006) derive new robust programming formulations enabling robust support vector machines for classification. Therefore they develop linear and second order cone programs and solve them with interior point methods.

Belacel et al. (2007) model the multicriteria fuzzy classification problem as nonlinear program and solve it using the RVNS metaheuristic. They show this method to outperform a number of traditional classifiers.

4. *Exploratory Data Analysis*—Most of the techniques for the DM task of exploratory data analysis aim at data visualization. Visualization of high volumes of high dimensional data records involves locating the records in a lower-dimensional space preserving the given distance measure.

Abbiw-Jackson et al. (2006) formulate the problem as a quadratic assignment problem and propose a divide and conquer local search heuristic for solution. Bernataviciene et al. (2006) realize visualization by a combination of self organizing maps and optimization based projection techniques.

5. *Discovering Patterns and Rules*—Rule induction methods and association rule algorithms are common techniques for the DM task of pattern and rule discovery. An example using optimization techniques well-known in OR is given by Baykasoglu and Özbakir (in press). They develop an evolutionary programming technique suitable for association rule mining. The method is shown to be very efficient and robust in comparison with standard DM algorithms.

Increased Effectiveness by Decision Structure Improvement

For some decision problems, nonlinearities or unknown system structure entail decision model structures not allowing for efficient search methods. In such cases, representation of decision variables is replaced by a higher order representation, e.g., by heuristic rules. If promising decision rules cannot be derived, DM is applied to achieve an improved representation.

Representation approaches in literature tend to be application domain specific. Therefore, they are classified according to the domains of manufacturing, logistics, warehousing, marketing, finance and health care management.

1. *Manufacturing*—Li and Olafsson (2005) present a DM approach to create decision rules for a single machine production scheduling environment. They apply a decision tree learning algorithm directly to production data to learn previously unknown dispatching rules. Huyet (2006) applies a decision tree algorithm to derive rules for a job shop composed of five workstations.

Shao et al. (2006) derive association rules for deciding on preferred product configuration alternatives given specific customer needs. Raheja et al. (2006) apply an association rule algorithm to derive rules for deciding on maintenance operations for systems in production environments.

2. *Warehousing*—Chen et al. (2005) and Wu (2006) provide two quite similar DM approaches. Both are shown to outperform commonly used heuristic decision rules.

Wu addresses the problem of selecting a subset of items to be stored in a special order completion zone. The number of items in the subset must not exceed a given threshold and the items must satisfy a given minimum percentage of orders. Wu transforms the problem into the well known DM problem of finding frequent itemsets. Application of a standard algorithm results in a ranking of itemsets.

Chen et al. address an order batching problem. They use an association rule algorithm to discover correlations between customer orders. Subsequently order batches are compiled according to the correlations with respect to the capacity constraints of storage/retrieval machines.

3. *Logistics*—Tseng et al. (2006) derive rules for supplier selection by a rough set procedure. Then they use a support vector machine learning procedure to increase rule accuracy. This method enables effective and timely supplier selection based on more features than lowest price only. Sawicki and Zak (2005) build an automatic deci-

sion making procedure for fleet maintenance. To this end, they derive maintenance rules using a rough set procedure.

4. *Marketing*—Cooper and Giuffrida (2000) decide on the amounts of stock needed for various products in retail stores. The number of units per product derived by a traditional market response model is adjusted according to rules derived by DM. The rule induction procedure considers attributes not taken into account by the market response model, hence includes more information into the decisions.
5. *Health Care Management*—Delesie and Croes (2000) apply a data visualization method to support multiple levels of decision making in the domain of health care management. A measure for similarity of hospitals represented by insurance reimbursements for a set of medical procedures is established. Then the hospitals are displayed according to mutual similarity. Finally the authors interpret the figures in order to derive decision guidelines.

Increased Effectiveness by Decision Model Improvement

DM can increase OR effectiveness by adaptation of a decision model allowing for efficient search methods. The following examples illustrate model adaptation in terms of parameters, decision variables and objective functions of well known decision models.

1. *Parameters*—Chen and Wu (2005) solve an order batching problem by a binary integer program. The aim is compiling batches to reduce travel distances in a warehouse. The decision variables indicate whether or not an order is assigned to a batch. An association rule algorithm is applied to characterize each pair of orders by a support value. This correlation measure is included as parameters into the decision model. Test results show reduced travel distance and a lower number of batches required compared to the results of standard batching procedures.
2. *Decision variables*—Brijs et al. (2004) determine the optimal product assortment for a retail store with respect to gross margin maximization. Given a maximum amount of handling and inventory costs, this can be modeled as a knapsack problem, i.e. as a binary integer program. An association rule algorithm is applied to determine sets of items offering cross-selling effects. Then, decision variables indicating whether a product set should be included into the assortment or not are added to the integer program. A branch and bound algorithm is applied to show the advantages of the approach.

Campbell et al. (2001) develop a decision model assigning the most profitable sequence of catalogs to each customer of a direct marketing company. However, optimizing over a set of 7 million customers is computationally infeasible. Hence, they choose an adequate representation of customers and form homogeneous customer groups by cluster analysis. Subsequently the optimization procedure is applied with customer groups instead of single customers as decision variables.

3. *Objective Function*—Bertsekas and Tsitsiklis (1996) introduce the neuro-dynamic methodology for adaptation of an objective function. The method provides an approximate dynamic programming approach suitable for sequential decision problems under uncertainty. The idea is determining the relation between the state of the optimized system and the cost-to-go value by application of a neural network to recorded system data.

FUTURE TRENDS

Increasing relevance of DM will encourage improved optimization approaches for all of the DM tasks and preprocessing. Further, economic developments create complex decision problems requiring elaborate decision models enabled by DM. DM-augmented OR approaches will emerge in the fields of supply chain management, as well as dynamic and stochastic vehicle routing and scheduling problems.

CONCLUSION

Three types of synergies of DM and OR are distinguished. The use of OR methods for increased efficiency of DM is the most established synergy. It results in an increasing number of significant improvements of DM

techniques. The use of DM for increased effectiveness by decision model structure improvement constitutes a younger field of research. A number of successful applications show that this synergy leads to an expansion of the OR problem domain. By decision model improvement DM enables application of traditional OR methods in complex environments. Though probably the most challenging of the three, this synergy is of high relevance with respect to a shifting focus of OR towards dynamic and stochastic problem environments.

REFERENCES

Abbiw-Jackson, R., Golden, B., Raghavan, S., & Wasil, E. (2006). A divide-and-conquer local search heuristic for data visualization, Computers & Operations Research, 33(11), 3070-3087.

Baykasoglu, A. & Özbakir, L. (in press). Mepar-miner: Multi-expression programming for classification rule mining, European Journal of Operational Research.

Belacel, N., Raval, H., & Punnen, A. (2007). Learning multicriteria fuzzy classification method proftan from data, Computers & Operations Research, 34(7), 1885-1898.

Beliakov, G., & King, M. (2006). Density based fuzzy c-means clustering of non-convex patterns, European Journal of Operational Research, 173(3), 717-728.

Bernataviciene, J., Dzemyda, G., Kurasova, O., & Marcinkevicius, V. (2006). Optimal decisions in combining the som with nonlinear projection methods, European Journal of Operational Research, 173(3), 729-745.

Bertsekas, D., & Tsitsiklis, J. (1996). Neuro-Dynamic Programming, Belmont, MA: Athena Scientific.

Boginski, V., Butenko, S., & Pardalos, P. (2006). Mining market data: A network approach, Computers & Operations Research, 33(11), 3171-3184.

Bradley, P., Fayyad, U., & Mangasarian, O. L. (1999). Mathematical programming for data mining, Journal of Computing, 11(3), 217-238.

Brijs, T., Swinnen, G., Vanhoof, K., & Wets, G. (2004). Building an association rules framework to improve product assortment decisions, Data Mining and Knowledge Discovery, 8(1), 7-23.

Campbell, D., Erdahl, R., Johnson, D., Bibelnieks, E., Haydock, M., Bullock, M., & Crowder, H. (2001). Optimizing customer mail streams at fingerhut, Interfaces, 31(1), 77-90.

Carrizosa, E., & Martin-Barragan, B. (2006). Two-group classification via a biobjective margin maximization model, European Journal of Operational Research, 173(3), 746-761.

Chen, M., Huang, C., Chen, K., & Wu, H. (2005). Aggregation of orders in distribution centers using data mining, Expert Systems with Applications, 28(3), 453-460.

Chen, M., & Wu, H. (2005). An association-based clustering approach to order batching considering customer demand patterns, Omega, 33(4), 333-343.

Cooper, L., & Giuffrida, G. (2000). Turning datamining into a management science tool: New algorithms and empirical results, Management Science, 46(2), 249-264.

Delesie, L., & Croes, L. (2000). Operations research and knowledge discovery: a data mining method applied to health care management, International Transactions in Operational Research, 7(2), 159-170.

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From data mining to knowledge discovery: An overview. In U. Fayyad (Ed.), Advances in Knowledge Discovery and Data Mining (pp. 1-34). Cambridge, MA: MIT Press.

Hand, D., Mannila, H., & Smyth, P. (2001). Principles of Data Mining, Cambridge, MA: MIT Press.

Hillier, F., & Lieberman, G. (2004). Introduction to Operations Research, New York, NY: McGraw-Hill.

Huyet, A. (2006). Optimization and analysis aid via data-mining for simulated production systems, European Journal of Operational Research, 173(3), 827-838.

Innis, T. (2006). Seasonal clustering technique for time series data, European Journal of Operational Research, 175(1), 376-384.

Janssens, D., Brijs, T., Vanhoof, K., & Wets, G. (2006). Evaluating the performance of cost-based discretization versus entropy- and error-based discretization, Computers & Operations Research, 33(11), 3107-3123.

Jones, D., Collins, A., & Hand, C. (2007). A classification model based on goal programming with non-standard preference functions with application to the prediction of cinema-going behaviour, European Journal of Operational Research, 177(1), 515-524.

Kulkarni, G., & Fathi, Y. (in press). Integer programming models for the q-mode problem, European Journal of Operational Research.

Li, X., & Olafsson, S. (2005). Discovering dispatching rules using data mining, Journal of Scheduling, 8(6), 515-527.

Mangasarian, O. L. (1997). Mathematical programming in data mining, Data Mining and Knowledge Discovery, 1(2), 183-201.

Meiri, R., & Zahavi, J. (2006). Using simulated annealing to optimize the feature selection problem in marketing applications, European Journal of Operational Research, 171(3), 842-858.

Meisel, S., & Mattfeld, D. (in press). Data Mining and Operations Research: Synergies of Integration, European Journal of Operational Research.

Padmanabhan, B., & Tuzhilin, A. (2003). On the use of optimization for data mining: Theoretical interactions and eCRM opportunities, Management Science, 49(10), 1327-1343.

Pendharkar, P. (2006). A data mining-constraint satisfaction optimization problem for cost effective classification, Computers & Operations Research, 33(11), 3124-3135.

Raheja, D., Linas, J., Nagi, R., & Romanowski, C. (2006). Data fusion/data mining-based architecture for condition-based maintenance, International Journal of Production Research, 44(14), 2869-2887.

Saglam, B., Salman, F., Saym, S., & Türkay, M. (2006). A mixed-integer programming approach to the clustering problem with an application in customer segmentation, European Journal of Operational Research, 173(3), 866-879.

Sawicki, P., & Zak, J. (2005). Technical diagnostic of a fleet of vehicles using rough sets theory, In Proceedings of the 16th Mini-EURO Conference, (pp. 770-779).

Shao, X., Wang, Z., Li, P., & Feng, C. (2006). Integrating data mining and rough set for customer group-based discovery of product configuration rules, International Journal of Production Research, 44(14), 2789-2811.

Spiliopoulos, K. & Sofianopoulou, S. (2007). Calculating distances for dissimilar strings: The shortest path formulation revisited, European Journal of Operational Research, 177(1), 525-539.

Trafalis, T., & Gilbert, R. (2006). Robust classification and regression using support vector machines, European Journal of Operational Research, 173(3), 893-909.

Tseng, J., Huang, C., Jiang, F., & Ho, J. (2006). Applying a hybrid data-mining approach to prediction problems: a case of preferred suppliers prediction, International Journal of Production Research, 44(14), 2935-2954.

Üney, F., & Türkay, M. (2006). A mixed-integer programming approach to multi-class data classification problem, European Journal of Operational Research, 173(3), 910-920.

White, J. (1991). An existence theorem for OR/MS, Operations Research, 39(2), 183-193.

Wu, C. (2006). Applying frequent itemset mining to identify a small itemset that satisfies a large percentage of orders in a warehouse, Computers & Operations Research, 33(11), 3161-3170.

Wu, X., Yu, P., Piatetsky-Shapiro, N., Cercone, N., Lin, T., Kotagiri, R., & Wah, B. (2003). Data Mining: How research meets practical development?, Knowledge and Information Systems, 5(2), 248-261.

Yang, J., & Olafsson, S. (2005). Intelligent partitioning for feature selection, Journal on Computing, 17(3), 339-355.

KEY TERMS

Decision Model: Formal statement expressing goal and feasible solutions of a decision problem.

Decision Model Structure: Generic specification of decision models expressing the model type as

well as restricting the algorithms suitable for problem solution.

Descriptive Modeling: Data Mining task aiming at development of models describing all the records in a given data set.

Discovering Patterns and Rules: Data Mining task aiming at discovery of particular aspects of a data set. These aspects are expressed as rules and patterns and refer to only some of the records in a given data set.

Exploratory Data Analysis: Data Mining task of data exploration without a clear goal of what is being looked for. Typically conducted by use of visual and interactive methods.

Predictive Modeling: Data Mining task aiming at development of models for prediction of the value of one attribute from known values of other attributes for each record in a given data set.

Preprocessing: Modification of the set of collected data records in order to create a target data set matching the requirements of the data mining task and algorithm considered subsequently.

Integration of Data Sources through Data Mining

Andreas Koeller
Montclair State University, USA

INTRODUCTION

Integration of data sources refers to the task of developing a common schema as well as data transformation solutions for a number of data sources with related content. The large number and size of modern data sources make manual approaches at integration increasingly impractical. Data mining can help to partially or fully automate the data integration process.

BACKGROUND

Many fields of business and research show a tremendous need to integrate data from different sources. The process of data source integration has two major components.

Schema matching refers to the task of identifying related fields across two or more databases (Rahm & Bernstein, 2001). Complications arise at several levels, for example

- Source databases can be organized by using several different models, such as the relational model, the object-oriented model, or semistructured models (e.g., XML).
- Information stored in a single table in one relational database can be stored in two or more tables in another. This problem is common when source databases show different levels of normalization and also occurs in nonrelational sources.
- A single field in one database, such as *Name*, could correspond to multiple fields, such as *First Name* and *Last Name*, in another.

Data transformation (sometimes called instance matching) is a second step in which data in matching fields must be translated into a common format. Frequent reasons for mismatched data include data format (such as *1.6.2004* vs. *6/1/2004*), numeric precision (*3.5kg* vs. *3.51kg*), abbreviations (*Corp.* vs. *Corporation*), or linguistic differences (e.g., using different synonyms for the same concept across databases).

Today's databases are large both in the number of records stored and in the number of fields (dimensions) for each datum object. Database integration or migration projects often deal with hundreds of tables and thousands of fields (Dasu, Johnson, Muthukrishnan, & Shkapenyuk, 2002), with some tables having 100 or more fields and/or hundreds of thousands of rows. Methods of improving the efficiency of integration projects, which still rely mostly on manual work (Kang & Naughton, 2003), are critical for the success of this important task.

MAIN THRUST

In this article, I explore the application of data-mining methods to the integration of data sources. Although data transformation tasks can sometimes be performed through data mining, such techniques are most useful in the context of schema matching. Therefore, the following discussion focuses on the use of data mining in schema matching, mentioning data transformation where appropriate.

Schema-Matching Approaches

Two classes of schema-matching solutions exist: schema-only-based matching and instance-based matching (Rahm & Bernstein, 2001).

Schema-only-based matching identifies related database fields by taking only the schema of input databases into account. The matching occurs through linguistic means or through constraint matching. Linguistic matching compares field names, finds similarities in field descriptions (if available), and attempts to match field names to names in a given hierarchy of terms *(ontology)*. Constraint matching matches fields based

on their domains (data types) or their key properties (primary key, foreign key). In both approaches, the data in the sources are ignored in making decisions on matching. Important projects implementing this approach include ARTEMIS (Castano, de Antonellis, & de Capitani di Vemercati, 2001) and Microsoft's CUPID (Madhavan, Bernstein, & Rahm, 2001).

Instance-based matching takes properties of the data into account as well. A very simple approach is to conclude that two fields are related if their minimum and maximum values and/or their average values are equal or similar. More sophisticated approaches consider the distribution of values in fields. A strong indicator of a relation between fields is a complete inclusion of the data of one field in another. I take a closer look at this pattern in the following section. Important instance-based matching projects are SemInt (Li & Clifton, 2000) and LSD (Doan, Domingos, & Halevy, 2001).

Some projects explore a combined approach, in which both schema-level and instance-level matching is performed. Halevy and Madhavan (2003) present a *Corpus-based* schema matcher. It attempts to perform schema matching by incorporating known schemas and previous matching results and to improve the matching result by taking such historical information into account.

Data-mining approaches are most useful in the context of instance-based matching. However, some mining-related techniques, such as graph matching, are employed in schema-only-based matching as well.

Instance-Based Matching through Inclusion Dependency Mining

An *inclusion dependency* is a pattern between two databases, stating that the values in a field (or set of fields) in one database form a subset of the values in some field (or set of fields) in another database. Such subsets are relevant to data integration for two reasons. First, fields that stand in an inclusion dependency to one another might represent related data. Second, knowledge of foreign keys is essential in successful schema matching. Because a foreign key is necessarily a subset of the corresponding key in another table, foreign keys can be discovered through inclusion dependency discovery.

The discovery of inclusion dependencies is a very complex process. In fact, the problem is in general NP-hard as a function of the number of fields in the largest inclusion dependency between two tables. However, a number of practical algorithms have been published.

De Marchi, Lopes, and Petit (2002) present an algorithm that adopts the idea of *levelwise discovery* used in the famous Apriori algorithm for association rule mining. Inclusion dependencies are discovered by first comparing single fields with one another and then combining matches into pairs of fields, continuing the process through triples, then 4-sets of fields, and so on. However, due to the exponential growth in the number of inclusion dependencies in larger tables, this approach does not scale beyond inclusion dependencies with a size of about eight fields.

A more recent algorithm (Koeller & Rundensteiner, 2003) takes a graph-theoretic approach. It avoids enumerating all inclusion dependencies between two tables and finds candidates for only the largest inclusion dependencies by mapping the discovery problem to a problem of discovering patterns (specifically cliques) in graphs. This approach is able to discover inclusion dependencies with several dozens of attributes in tables with tens of thousands of rows. Both algorithms rely on the antimonotonic property of the inclusion dependency discovery problem. This property is also used in association rule mining and states that patterns of size k can only exist in the solution of the problem if certain patterns of sizes smaller than k exist as well. Therefore, it is meaningful to first discover small patterns (e.g., single-attribute inclusion dependency) and use this information to restrict the search space for larger patterns.

Instance-Based Matching in the Presence of Data Mismatches

Inclusion dependency discovery captures only part of the problem of schema matching, because only *exact* matches are found. If attributes across two relations are not exact subsets of each other (e.g., due to entry errors), then data mismatches requiring data transformation, or partially overlapping data sets, it becomes more difficult to perform data-driven mining-based discovery. Both false negatives and false positives are possible. For example, matching fields might not be discovered due to different encoding schemes (e.g., use of a numeric identifier in one table, where text is used to denote the same values in another table). On the other hand, purely data-driven discovery relies on the assumption that semantically related values are

also syntactically equal. Consequently, fields that are discovered by a mining algorithm to be matching might not be semantically related.

Data Mining by Using Database Statistics

The problem of false negatives in mining for schema matching can be addressed by more sophisticated mining approaches. If it is known which attributes across two relations relate to one another, *data transformation* solutions can be used. However, automatic discovery of matching attributes is also possible, usually through the evaluation of statistical patterns in the data sources. In the classification of Kang and Naughton (2003), interpreted matching uses artificial intelligence techniques, such as Bayesian classification or neural networks, to establish hypotheses about related attributes. In the uninterpreted matching approach, statistical features, such as the unique value count of an attribute or its frequency distribution, are taken into consideration. The underlying assumption is that two attributes showing a similar distribution of unique values might be related even though the actual data values are not equal or similar.

Another approach for detecting a semantic relationship between attributes is to use information entropy measures. In this approach, the concept of *mutual information*, which is based on entropy and conditional entropy of the underlying attributes, "measures the reduction in uncertainty of one attribute due to the knowledge of the other attribute" (Kang & Naughton, 2003, p. 207).

Further Problems in Information Integration through Data Mining

In addition to the approaches mentioned previously, several other data-mining and machine-learning approaches, in particular classification and rule-mining techniques, are used to solve special problems in information integration.

For example, a common problem occurring in real-world integration projects is related to duplicate records across two databases, which must be identified. This problem is usually referred to as the *record-linking problem* or the *merge/purge problem* (Hernandéz & Stolfo, 1998). Similar statistical techniques as the ones described previously are used to approach this problem. The Commonwealth Scientific and Industrial Research Organisation (2003) gives an overview of approaches, which include decision models, predictive models such as support vector machines, and a Bayesian decision cost model.

In a similar context, data-mining and machine-learning solutions are used to improve the data quality of existing databases as well. This important process is sometimes called *data scrubbing*. Lübbers, Grimmer, and Jarke (2003) present a study of the use of such techniques and refer to the use of data mining in data quality improvement as *data auditing*.

Recently, the emergence of *Web Services* such as XML, SOAP, and UDDI promises to open opportunities for database integration. Hansen, Madnick, and Siegel (2002) argue that Web services help to overcome some of the technical difficulties of data integration, which mostly stem from the fact that traditional databases are not built with integration in mind. On the other hand, Web services by design standardize data exchange protocols and mechanisms. However, the problem of identifying semantically related databases and achieving schema matching remains.

FUTURE TRENDS

Increasing amounts of data are being collected at all levels of business, industry, and science. Integration of data also becomes more and more important as businesses merge and research projects increasingly require interdisciplinary efforts. Evidence for the need for solutions in this area is provided by the multitude of partial software solutions for such business applications as ETL (Pervasive Software, Inc., 2003), and by the increasing number of integration projects in the life sciences, such as Genbank by the National Center for Biotechnology Information (NCBI) or Gramene by the Cold Spring Harbor Laboratory and Cornell University. Currently, the integration of data sources is a daunting task, requiring substantial human resources. If automatic methods for schema matching were more readily available, data integration projects could be completed much faster and could incorporate many more databases than is currently the case.

Furthermore, an emerging trend in data source integration is the move from batch-style integration, where a set of given data sources is integrated at one time into one system, to real-time integration, where

data sources are immediately added to an integration system as they become available. Solutions to this new challenge can also benefit tremendously from semiautomatic or automatic methods of identifying database structure and relationships.

CONCLUSION

Information integration is an important and difficult task for businesses and research institutions. Although data sources can be integrated with each other by manual means, this approach is not very efficient and does not scale to the current requirements. Thousands of databases with the potential for integration exist in every field of business and research, and many of those databases have a prohibitively high number of fields and/or records to make manual integration feasible. Semiautomatic or automatic approaches to integration are needed.

Data mining provides very useful tools to automatic data integration. Mining algorithms are used to identify schema elements in unknown source databases, to relate those elements to each other, and to perform additional tasks, such as data transformation. Essential business tasks such as extraction, transformation, and loading (ETL) and data integration and migration in general become more feasible when automatic methods are used.

Although the underlying algorithmic problems are difficult and often show exponential complexity, several interesting solutions to the schema-matching and data transformation problems in integration have been proposed. This is an active area of research, and more comprehensive and beneficial applications of data mining to integration are likely to emerge in the near future.

REFERENCES

Castano, S., de Antonellis, V., & de Capitani di Vemercati, S. (2001). Global viewing of heterogeneous data sources. *IEEE Transactions on Knowledge and Data Engineering, 13*(2), 277-297.

Commonwealth Scientific and Industrial Research Organisation. (2003, April). *Record linkage: Current practice and future directions* (CMIS Tech. Rep. No. 03/83). Canberra, Australia: L. Gu, R. Baxter, D. Vickers, & C. Rainsford. Retrieved July 22, 2004, from *http://www.act.cmis.csiro.au/rohanb/PAPERS/record_linkage.pdf*

Dasu, T., Johnson, T., Muthukrishnan, S., & Shkapenyuk, V. (2002). Mining database structure; or, how to build a data quality browser. *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data*, USA (pp. 240-251).

de Marchi, F., Lopes, S., & Petit, J.-M. (2002). Efficient algorithms for mining inclusion dependencies. *Proceedings of the Eighth International Conference on Extending Database Technology,* Prague, Czech Republic, 2287 (pp. 464-476).

Doan, A. H., Domingos, P., & Halevy, A. Y. (2001). Reconciling schemas of disparate data sources: A machine-learning approach. *Proceedings of the ACM SIGMOD International Conference on Management of Data,* USA (pp. 509-520).

Halevy, A. Y., & Madhavan, J. (2003). Corpus-based knowledge representation. *Proceedings of the 18th International Joint Conference on Artificial Intelligence,* Mexico (pp. 1567-1572).

Hernández, M. A., & Stolfo, S. J. (1998). Real-world data is dirty: Data cleansing and the merge/purge problem. *Journal of Data Mining and Knowledge Discovery, 2*(1), 9-37.

Kang, J., & Naughton, J. F. (2003). On schema matching with opaque column names and data values. *Proceedings of the ACM SIGMOD International Conference on Management of Data,* USA (pp. 205-216).

Koeller, A., & Rundensteiner, E. A. (2003). Discovery of high-dimensional inclusion dependencies. *Proceedings of the 19th IEEE International Conference on Data Engineering,* India (pp. 683-685).

Li, W., & Clifton, C. (2000). SemInt: A tool for identifying attribute correspondences in heterogeneous databases using neural network. *Journal of Data and Knowledge Engineering, 33*(1), 49-84.

Lübbers, D., Grimmer, U., & Jarke, M. (2003). Systematic development of data mining-based data quality tools. *Proceedings of the 29th International Conference on Very Large Databases,* Germany (pp. 548-559).

Madhavan, J., Bernstein, P. A., & Rahm, E. (2001) Generic schema matching with CUPID. *Proceedings of the 27th International Conference on Very Large Databases,* Italy (pp. 49-58).

Massachusetts Institute of Technology, Sloan School of Management. (2002, May). *Data integration using Web services* (Working Paper 4406-02). Cambridge, MA. M. Hansen, S. Madnick, & M. Siegel. Retrieved July 22, 2004, from *http://hdl.handle.net/1721.1/1822*

Pervasive Software, Inc. (2003). *ETL: The secret weapon in data warehousing and business intelligence.* [Whitepaper]. Austin, TX: Pervasive Software.

Rahm, E., & Bernstein, P. A. (2001). A survey of approaches to automatic schema matching. *VLDB Journal, 10*(4), 334-350.

KEY TERMS

Antimonotonic: A property of some pattern-finding problems stating that patterns of size *k* can only exist if certain patterns with sizes smaller than *k* exist in the same dataset. This property is used in levelwise algorithms, such as the Apriori algorithm used for association rule mining or some algorithms for inclusion dependency mining.

Database Schema: A set of names and conditions that describe the structure of a database. For example, in a relational database, the schema includes elements such as table names, field names, field data types, primary key constraints, or foreign key constraints.

Domain: The set of permitted values for a field in a database, defined during database design. The actual data in a field are a subset of the field's domain.

Extraction, Transformation, and Loading (ETL): Describes the three essential steps in the process of data source integration: extracting data and schema from the sources, transforming it into a common format, and loading the data into an integration database.

Foreign Key: A key is a field or set of fields in a relational database table that has unique values, that is, no duplicates. A field or set of fields whose values form a subset of the values in the key of *another* table is called a foreign key. Foreign keys express relationships between fields of different tables.

Inclusion Dependency: A pattern between two databases, stating that the values in a field (or set of fields) in one database form a subset of the values in some field (or set of fields) in another database.

Levelwise Discovery: A class of data-mining algorithms that discovers patterns of a certain size by first discovering patterns of size 1, then using information from that step to discover patterns of size 2, and so on. A well-known example of a levelwise algorithm is the Apriori algorithm used to mine association rules.

Merge/Purge: The process of identifying duplicate records during the integration of data sources. Related data sources often contain overlapping information extents, which have to be reconciled to improve the quality of an integrated database.

Relational Database: A database that stores data in *tables,* which are sets of tuples (rows). A set of corresponding values across all rows of a table is called an *attribute, field,* or *column.*

Schema Matching: The process of identifying an appropriate mapping from the schema of an input data source to the schema of an integrated database.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 625-629, copyright 2005 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

Integrative Data Analysis for Biological Discovery

Sai Moturu
Arizona State University, USA

Lance Parsons
Arizona State University, USA

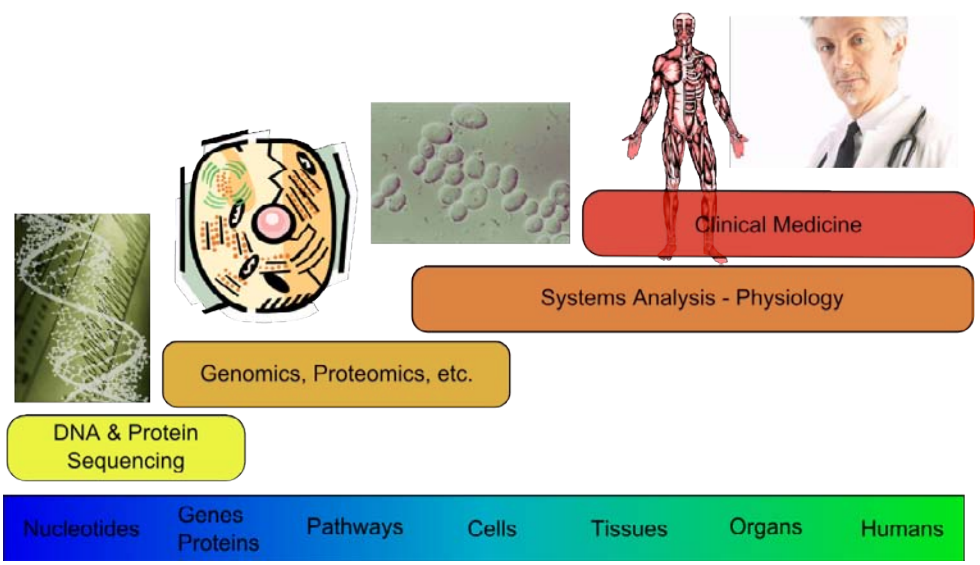
Zheng Zhao
Arizona State University, USA

Huan Liu
Arizona State University, USA

INTRODUCTION

As John Muir noted, "When we try to pick out anything by itself, we find it hitched to everything else in the Universe" (Muir, 1911). In tune with Muir's elegantly stated notion, research in molecular biology is progressing toward a systems level approach, with a goal of modeling biological systems at the molecular level. To achieve such a lofty goal, the analysis of multiple datasets is required to form a clearer picture of entire biological systems (Figure 1). Traditional molecular biology studies focus on a specific process in a complex biological system. The availability of high-throughput technologies allows us to sample tens of thousands of features of biological samples at the molecular level. Even so, these are limited to one particular view of a biological system governed by complex relationships and feedback mechanisms on a variety of levels. Integrated analysis of varied biological datasets from the genetic, translational, and protein levels promises more accurate and comprehensive results, which help discover concepts that cannot be found through separate, independent analyses. With this article, we attempt to provide a comprehensive review of the existing body of research in this domain.

Figure 1. Complexity increases from the molecular and genetic level to the systems level view of the organism (Poste, 2005).



BACKGROUND

The rapid development of high-throughput technologies has allowed biologists to obtain increasingly comprehensive views of biological samples at the genetic level. For example, microarrays can measure gene expression for the complete human genome in a single pass. The output from such analyses is generally a list of genes (features) that are differentially expressed (upregulated or downregulated) between two groups of samples or ones that are coexpressed across a group of samples. Though every gene is measured, many are irrelevant to the phenomenon being studied. Such irrelevant features tend to mask interesting patterns, making gene selection difficult. To overcome this, external information is required to draw meaningful inferences (guided feature selection). Currently, numerous high-throughput techniques exist along with diverse annotation datasets presenting considerable challenges for data mining (Allison, Cui, Page & Sabripour, 2006).

Sources of background knowledge available include metabolic and regulatory pathways, gene ontologies, protein localization, transcription factor binding, molecular interactions, protein family and phylogenetic information, and information mined from biomedical literature. Sources of high-throughput data include gene expression microarrays, comparative genomic hybridization (CGH) arrays, single nucleotide polymorphism (SNP) arrays, genetic and physical interactions (affinity precipitation, two-hybrid techniques, synthetic lethality, synthetic rescue) and protein arrays (Troyanskaya, 2005). Each type of data can be richly annotated using clinical data from patients and background knowledge. This article focuses on studies using microarray data for the core analysis combined with other data or background knowledge. This is the most commonly available data at the moment, but the concepts can be applied to new types of data and knowledge that will emerge in the future.

Gene expression data has been widely utilized to study varied things ranging from biological processes and diseases to tumor classification and drug discovery (Carmona-Saez, Chagoyen, Rodriguez, Trelles, Carazo & Pascual-Montano, 2006). These datasets contain information for thousands of genes. However, due to the high cost of these experiments, there are very few samples relative to the thousands of genes. This leads to the curse of dimensionality (Yu & Liu 2004). Let M be the number of samples and N be the number of genes. Computational learning theory suggests that the search space is exponentially large in terms of N and the required number of samples for reliable learning about given phenotypes is on the scale of $O(2^N)$ (Mitchell, 1997; Russell & Norvig, 2002). However, even the minimum requirement (M=10*N) as a statistical "rule of thumb" is patently impractical for such a dataset (Hastie, Tibshirani & Friedman, 2001). With limited samples, analyzing a dataset using a single criterion leads to the selection of many statistically relevant genes that are equally valid in interpreting the data. However, it is commonly observed that statistical significance may not always correspond to biological relevance. Traditionally, additional information is used to guide the selection of biologically relevant genes from the list of statistically significant genes. Using such information during the analysis phase to guide the mining process is more effective, especially when dealing with such complex processes (Liu, Dougherty, Dy, Torkkola, Tuv, Peng, Ding, Long, Berens, Parsons, Zhao, Yu & Forman, 2005; Anastassiou, 2007).

Data integration has been studied for a long time, ranging from early applications in distributed databases (Deen, Amin & Taylor, 1987) to the more recent ones in sensor networks (Qi, Iyengar & Chakrabarty, 2001) and even biological data (Lacroix, 2002; Searls, 2003). However, the techniques we discuss are those using integrative analyses as opposed to those which integrate data or analyze such integrated data. The difference lies in the use of data from multiple sources in an integrated analysis framework. The range of biological data available and the variety of applications make such analyses particularly necessary to gain biological insight from a whole organism perspective.

MAIN FOCUS

With the increase in availability of several types of data, more researchers are attempting integrated analyses. One reason for using numerous datasets is that high-throughput data often sacrifice specificity for scale (Troyanskaya, 2005), resulting in noisy data that might generate inaccurate hypotheses. Replication of experiments can help remove noise but they are costly. Combining data from different experimental sources and knowledge bases is an effective way to reduce the effects of noise and generate more accurate hypotheses. Multiple sources provide additional information that

when analyzed together, can explain some phenomenon that one source cannot. This is commonly observed in data mining and statistical pattern recognition problems (Troyanskaya, 2005; Rhodes & Chinnaiyan, 2005; Quackenbush, 2007). In addition, the systems being studied are inherently complex (Figure 1). One dataset provides only one view of the system, even if it looks at it from a whole organism perspective. As a result, multiple datasets providing different views of the same system are intuitively more useful in generating biological hypotheses.

There are many possible implementation strategies to realize the goals of integrative analysis. One can visualize the use of one dataset after another in a serial model or all the datasets together in a parallel model. A serial model represents classical analysis approaches (Figure 2a). Using all the datasets at once to perform an integrative analysis would be a fully parallel model (Figure 2b). By combining those approaches, we arrive at a semi-parallel model that combines some of the data early on, but adds some in later stages of the analysis. Such an analysis could integrate different groups of data separately and then integratively process the results (Figure 2c).

The fully parallel model combines all the datasets together and performs the analysis without much user intervention. Any application of expert knowledge is done at the preprocessing stage or after the analysis. An algorithm that follows this model allows the use of heterogeneous datasets and can therefore be applied in later studies without need for modification.

The semi-parallel model is used when expert knowledge is needed to guide the analysis towards more specific goals. A model that can be tailored to a study and uses expert knowledge can be more useful than a generic model. An algorithm that follows this model requires tuning specific to each experiment and cannot be applied blindly to diverse datasets. These models represent the basic ways of performing an integrative analysis.

REPRESENTATIVE SELECTION OF ALGORITHMS

Integrative analyses using data from diverse sources have shown promise of uncovering knowledge that is not found when analyzing a single dataset. These studies analyze disparate datasets to study subjects ranging from gene function to protein-protein interaction. Common to all these studies is the use of expert knowledge and an increase in prediction accuracy when executing integrative analysis of varied datasets compared to results obtained when executing classical analysis with a single data source. More specifically these studies use data sources that include both high-throughput and conventional data to perform an analysis and generate inferences. This is in contrast to conventional studies where a high-throughput data source is used for analysis and additional information is later sought to help in drawing inferences. A variety of data mining techniques including decision trees, biclustering, Bayesian networks, kernel methods have been applied to the integrated analysis of biological data.

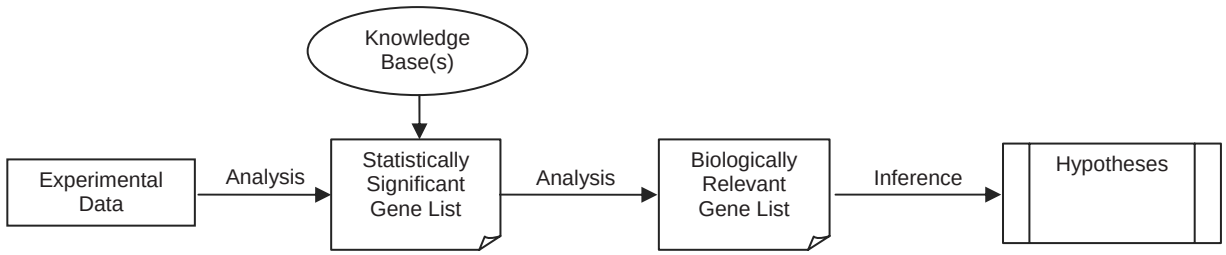
Troyanskaya, Dolinksi, Owen, Altman & Botstein (2003) developed a framework called Multisource Association of Genes by Integration of Clusters (MAGIC) that allows the combination of different biological data with expression information to make predictions about gene function using Bayesian reasoning. The structure of the Bayesian network is created using expert input. The inputs to this network are matrices representing gene-gene relationships resulting in groups of functionally related genes as output. This approach would fall under the fully parallel model with the use of expert knowledge at the preprocessing stage. The data is processed individually to create matrices and all these matrices are analyzed together integratively.

Jansen, Yu, Greenbaum, Kluger, Krogan, Chung, Emili, Snyder, Greenblatt & Gerstein (2003) predicted protein-protein interactions using experimental and annotation data separately. Bayesian networks were used to obtain an experimental probabilistic interactome and a predicted probabilistic interactome respectively. These two were then combined to give a final probabilistic interactome. This experiment would fall under the semi-parallel model.

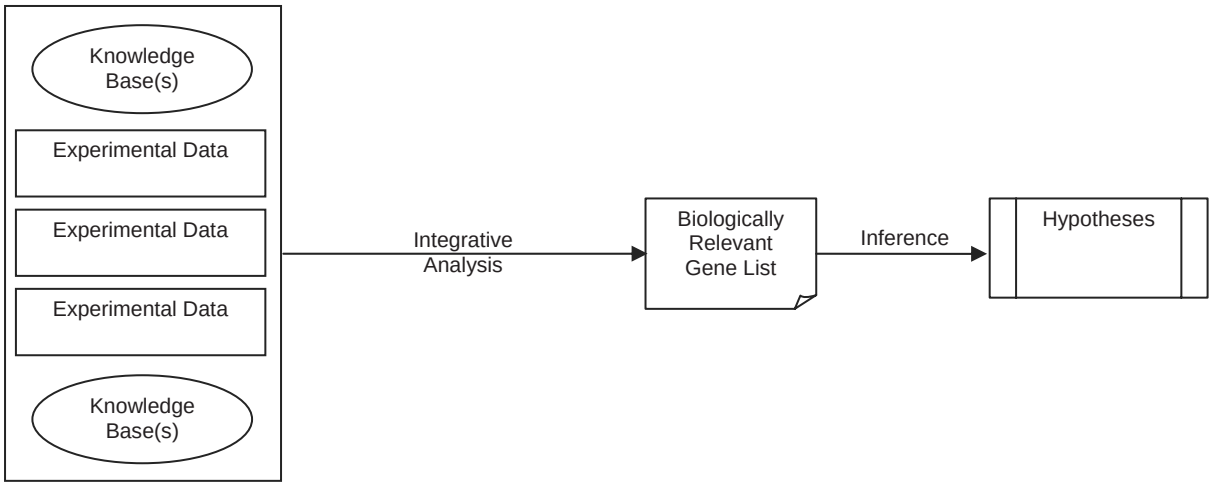
Jiang & Keating (2003) developed a multistage learning framework called Annotation Via Integration of Data (AVID) for prediction of functional relationships among proteins. High-confidence networks are built in which proteins are connected if they are likely to share a common annotation. Functional annotation is treated as a classification problem and annotations are assigned to unannotated proteins based on their neighbors in the network. This method also falls into the fully parallel model but in a multi stage implementation.

Figure 2. Integrative Analysis Models: a. Serial, b. Fully Parallel, c. Semi-Parallel

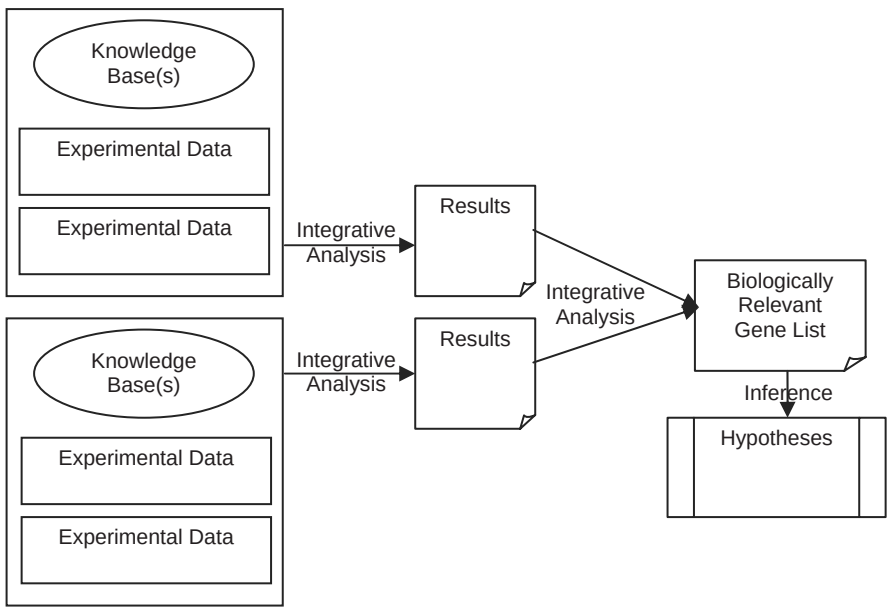
a. Model for analysis of a single high-throughput dataset with the aid of knowledge base(s) (serial analysis)



b. Model for Integrative analysis of multiple high-throughput datasets with the aid of knowledge base(s) (fully parallel analysis)



c. Model for Integrative analysis of multiple high-throughput datasets with the aid of knowledge base(s) (semi-parallel analysis)



Tanay, Sharan, Kupiec & Shamir (2004) proposed Statistical-Algorithmic Method for Bicluster Analysis (SAMBA) to analyze mixed datasets to identify genes that show statistically significant correlation across these sources. The data is viewed as properties of the genes and the underlying biclustering algorithm looks for statistically significant subgraphs from the genes-properties graph to predict functional groupings for various genes. This method again falls under the fully parallel model with some preprocessing.

Lanckriet, Bie, Cristianini, Jordan & Noble (2004) proposed a computational framework where each dataset is represented using a kernel function and these kernel functions are later combined to develop a learning algorithm to recognize and classify yeast ribosomal and membrane proteins. This method also could be categorized as a fully parallel model.

Zhang, Wong, King & Roth (2004) used a probabilistic decision tree to analyze protein-protein interactions from high-throughput methods like yeast two-hybrid and affinity purification coupled with mass spectrometry along with several other protein-pair characteristics to predict co-complexed pairs of proteins. This method again follows a fully parallel model with all the datasets contributing to the analysis.

Carmona-Saez, Chagoyen, Rodriguez, Trelles, Carazo, & Pascual-Montano (2006) developed a method for integrated analysis of gene expression data using association rules discovery. The itemsets used include the genes along with the set of experiments in which the gene is differentially expressed and the gene annotation. The generated rules are constrained to have gene annotation as an antecedent. Since the datasets used were well studied, the generated rules could be compared with the known information to find that they really are insightful.

APPLICATION TO A SPECIFIC DOMAIN: CANCER

Cancer studies provide a particularly interesting application of integrated analysis techniques. Different datasets from studies interrogating common hypotheses can be used to perform meta-analyses enabling the identification of robust gene expression patterns. These patterns could be identified in a single type or across different cancers. Annotation databases and pathway resources can be used for functional enrichment of cancer signatures by reducing large signatures to a smaller number of genes. Such additional knowledge is useful to view the existing information in a new light. Understanding a complex biological process requires us to view such knowledge in terms of complex molecular networks. To understand such networks, protein-protein interactions need to be mapped. Information on such protein interaction networks is limited.

In addition to protein interaction networks, global transcriptional networks can be used to enhance our interpretation of cancer signatures. Knowledge of transcription factor binding site information allows us to identify those that might be involved in particular signatures. Studying model oncogene systems can further enhance our understanding of these signatures (Rhodes & Chinnaiyan, 2005; Hu, Bader, Wigle & Emili, 2007). Examples of studies employing these approaches show the usefulness of integrative analyses in general as well as for their application to cancer.

FUTURE TRENDS

The future trends in this area can be divided into three broad categories: experimental techniques, analysis methods and application areas. Though, we are more interested in the analysis methods, these categories are interlinked and drive each other in terms of change and growth. In addition to existing high-throughput techniques, advancement in newer approaches like SNP arrays, array comparative genome hybridization, promoter arrays, proteomics and metabolomics would provide more high quality information to help us understand complex biological processes (Rhodes & Chinnaiyan, 2005). With such an aggregation of data, integrative analyses techniques would be a key to gaining maximum understanding of these processes.

Apart from experimental techniques, there is tremendous unrealized potential in integrative analysis techniques. A majority of the current methods are fully parallel and generic. They can be applied to different studies using dissimilar datasets but asking the same biological question. Though domain knowledge guides the development of these algorithms to an extent, optimal use of such knowledge is not seen. With the huge amount of diversity in the datasets, the use of expert knowledge is needed to develop a method specific to a domain based on the types of data being used and the questions being asked. This means alternate semi-

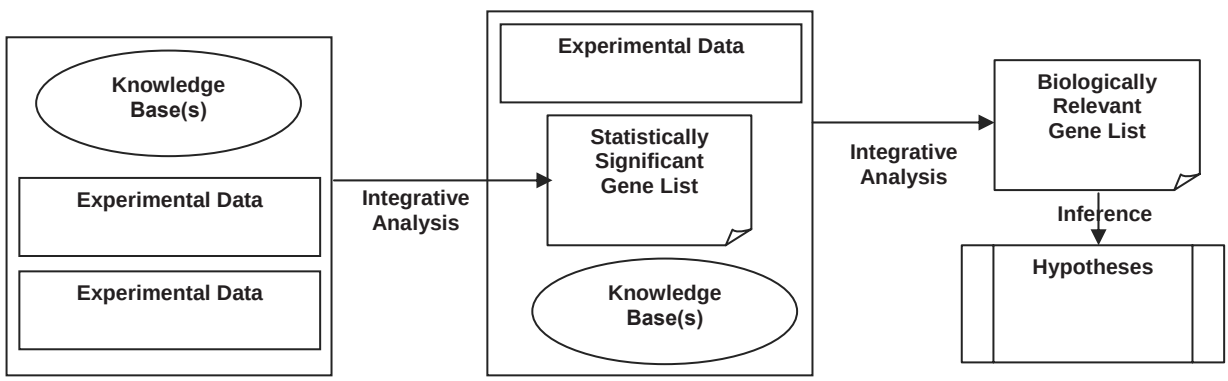
parallel integrative analysis techniques might be seen in the future that are better suited to a specific study. Such methods can then be adapted to other studies using knowledge about that particular domain and those datasets.

Apart from the semi-parallel analysis model seen earlier (Figure 2c), there can be other variations of the model. One such model (Figure 3a) uses few datasets together to perform the analysis and uses background knowledge (left out of the analysis phase intentionally based on the nature of the study or unintentionally based on the nature of the data) to further refine hypotheses. Yet another variation of the semi-parallel model (Figure 3b) analyzes some of these datasets individually and uses the generated results to guide the integrative analysis of the other datasets. An example would be to use the results for each gene from an analysis as annotation while analyzing the other datasets. Though many techniques for integrative analysis are seen, a comprehensive evaluation methodology still needs to be adopted to allow for comparisons between techniques and evaluation across varying datasets.

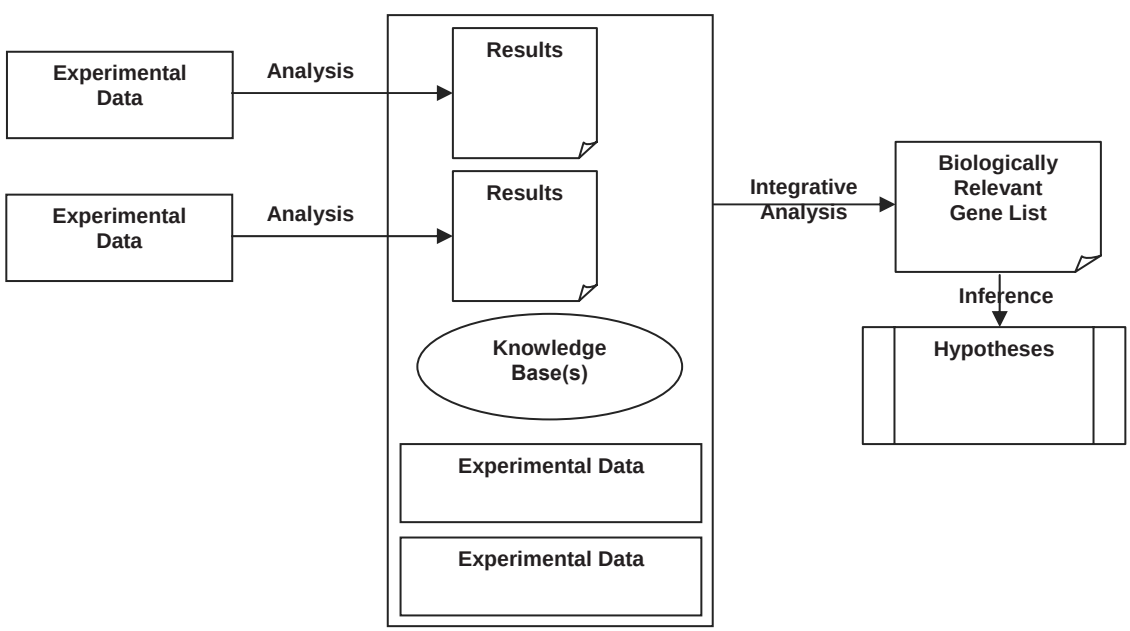
In addition, there is room to expand the application of these techniques in the current domains as well

Figure 3. Alternate semi-parallel analysis models

a. Alternate model 1 for semi-parallel integrative analysis of multiple high-throughput datasets with the aid of knowledge base(s)



b. Alternate model 2 for semi-parallel integrative analysis of multiple high-throughput datasets with the aid of knowledge base(s)



as to apply them to newer domains like personalized medicine. In most of the existing studies, the prominent datasets have been at the gene and protein expression level. In the future, we can foresee other datasets like SNP data and clinical data contributing equally to biological discovery.

CONCLUSION

The field of biological and biomedical informatics has been developing at a searing pace in recent years. This has resulted in the generation of a voluminous amount of data waiting to be analyzed. The field of data mining has been up to the challenge so far, with researchers taking a keen interest and providing impressive solutions to research questions. However, with the generation of heterogeneous datasets, the scope for analysis from a systems biology perspective has increased. This has stimulated the move from conventional independent analyses to novel integrated analyses.

Integrative analysis of biological data has been employed for a short while with some success. Different techniques for such integrative analyses have been developed. We have tried to summarize these techniques in the framework of the models described in Figure 2 and Figure 3. However, varied research questions and applications still exist and new data is being created continuously. Hence, there is a lot of scope for improvement and development of new techniques in this fledgling area, which could ultimately lead to important biological discoveries.

REFERENCES

Allison, D. B., Cui, X., Page, G. P, & Sabripour, M. (2006). Microarray data analysis: from disarray to consolidation and consensus. *Nat Rev Genet, 7(1),* 55-65.

Anastassiou, D. (2007). Computational analysis of the synergy among multiple interacting genes. *Molecular Systems Biology, 3*, 83.

Carmona-Saez, P., Chagoyen, M., Rodriguez, A., Trelles, O., Carazo, J. M., & Pascual-Montano, A. (2006). Integrated analysis of gene expression by association rules discovery. *BMC Bioinformatics, 7*, 54.

Deen, S. M., Amin, R. R., & Taylor, C. C. (1987). Data integration in distributed databases. *IEEE Transactions on Software Engineering, 13(7)*, 860-864.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The Elements of Statistical Learning*. New York, NY: Springer.

Hu, P., Bader, G., Wigle, D. A., & Emili, A. (2007). Computational prediction of cancer-gene function. *Nature Reviews Cancer, 7*, 23-34.

Jansen, R., Yu, H., Greenbaum, D., Kluger, Y., Krogan, N. J., Chung, S., Emili, A., Snyder, M., Greenblatt, J. F., & Gerstein, M. (2003). A Bayesian Networks Approach for Predicting Protein-Protein Interactions from Genomic Data. *Science, 302(5644)*, 449-453.

Jiang, T., & Keating, A. E. (2005). AVID: An integrative framework for discovering functional relationships among proteins. *BMC Bioinformatics, 6(1),* 136.

Lacroix, Z. (2002). Biological Data Integration: Wrapping data & tools. *IEEE Transactions on Information Technology in Biomedicine, 6(2),* 123-128.

Lanckriet, G. R. G., Bie, T. D., Cristianini, N., Jordan, M. I., & Noble, W. S. (2004). A statistical framework for genomic data fusion. *Bioinformatics, 20(16),* 2626-2635.

Liu, H., Dougherty, E. R., Dy, J. G., Torkkola, K., Tuv, E., Peng, H., Ding, C., Long, F., Berens, M., Parsons, L., Zhao, Z., Yu, L., & Forman, G. (2005). Evolving feature selection. *IEEE Intelligent Systems, 20(6)*, 64-76.

Mitchell, T. M. (1997). *Machine Learning.* New York, NY: McGraw Hill.

Muir, J. (1911). *My First Summer in the Sierra*. Boston, MA: Houghton Mifflin.

Poste, G. (2005, November). Integrated Biosystems Research. *Biodesign Institute Fall Workshop.* Retrieved March 30, 2006, from http://www.biodesign.asu.edu/news/99/

Qi, H., Iyengar, S. S., & Chakrabarty, K. (2001). Multiresolution data integration using mobile agents in distributed sensor networks. *IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews, 31(3),* 383-391.

Quackenbush, J. (2007). Extracting biology from high-dimensional biological data. *The Journal of Experimental Biology, 210*, 1507-1517.

Rhodes, D.R., & Chinnaiyan, A.M. (2005). Integrative analysis of the cancer transcriptome. *Nature Genetics, 37*, S31-S37.

Russell, S. J., & Norvig, P. (2002). *Artificial Intelligence: A Modern Approach*. Upper Saddle River, NJ: Prentice Hall.

Searls, D. B. (2003). Data integration-connecting the dots. *Nature Biotechnology, 21(8)*, 844-845.

Tanay, A., Sharan, R., Kupiec, M., & Shamir, R. (2004). Revealing modularity and organization in the yeast molecular network by integrated analysis of highly heterogeneous genomewide data. *Proc Natl Acad Sci U S A, 101(9)*, 2981-2986.

Troyanskaya, O. G., Dolinksi, K., Owen, A. B., Altman, R. B., & Botstein, D. (2003). A Bayesian framework for combining heterogeneous data sources for gene function prediction (in Saccharomyces cerevisiae). *Proc Natl Acad Sci U S A, 100(14)*, 8348-8353.

Troyanskaya, O. G. (2005). Putting microarrays in a context: Integrated analysis of diverse biological data. *Briefings in Bioinformatics, 6(1)*, 34-43.

Yu, L., & Liu, H. (2004). Efficient Feature Selection via Analysis of Relevance and Redundancy. *Journal of Machine Learning, 5*, 1205-1224.

Zhang, L. V., Wong, S. L., King, O. D., & Roth, F. P. (2004). Predicting co-complexed protein pairs using genomic and proteomic data integration. *BMC Bioinformatics, 5*, 38.

KEY TERMS

Annotation: Additional information that helps understand and interpret data better.

Biological Relevance: Relevance from a biological perspective as opposed to a statistical perspective.

Data Integration: Integration of multiple datasets from disparate sources.

DNA Microarrays: A technology used to measure the gene expression levels of thousands of genes at once.

Genomics/Proteomics: The study of the complete collection of knowledge encoded by DNA and proteins, respectively, in an organism.

High-throughput techniques: Biological techniques capable of performing highly parallel analyses that generate a large amount of data through a single experiment.

Integrated/Integrative Data Analysis: The analysis of multiple heterogeneous datasets under an integrated framework suitable to a particular application.

Meta-analyses: The combination and analysis of results from multiple independent analyses focusing on similar research questions.

Intelligent Image Archival and Retrieval System

P. Punitha
University of Glasgow, UK

D. S. Guru
University of Mysore, India

INTRODUCTION

'A visual idea is more powerful than verbal idea', 'A picture is worth more than ten thousand words', 'No words can convey what a picture speaks', 'A picture has to be seen and searched as a picture only' are few of the well-known sayings that imply the certainty for the widespread availability of images. Common sense evidence suggests that images are required for a variety of reasons, like, illustration of text articles, conveying information or emotions that are difficult to describe in words, display of detailed data for analysis (medical images), formal recording of design data for later use (architectural plans) etc.

The advent of digital photography combined with decreasing storage and processing cost, allows more and more people to have their personal collection of photographs and other visual content available on the internet. Organising these digital images into a small number of categories and providing effective indexing is imperative for accessing, browsing and retrieving useful data in "real time". The process of digitization does not in itself make image collections easier to manage. Some form of indexing (cataloguing) is still necessary. People's interest to have their own digital libraries has burgeoned and hence requires a data structure to preserve the images for a long time and also provide easy access to the desired images. These requirements have indeed forced the design of specialized imaging systems/ image databases, such that an access to any image is effective and efficient.

An efficient image archival and retrieval system is characterized by its ability to retrieve relevant images based on their visual and semantic contents rather than using simple attributes or keywords assigned to them. Thus, it is necessary to support queries based on image semantics rather than mere-pixel-to-pixel matching. An image archival and retrieval system should therefore allow adequate abstraction mechanisms for capturing higher level semantics of images in order to support content addressability as far as possible. That is, for two images to be similar, not only the shape, color and texture properties of individual image regions must be similar, but also they must have the same arrangement (i.e., spatial relationships) in both the images. In fact, this is the strategy, which is generally being employed by our vision system most of the times. An effective method of representing images depends on the perception of knowledge embedded in images in terms of objects/components (generally known as elements) present in them along with their topological relationships. The perception of topological relationships, especially spatial relationships existing among the significant elements of an image, helps in making the image database system more intelligent, fast and flexible.

An obvious method to search an image database is sequential scanning. The query is matched with all stored images (i.e., the representation of the query is matched with all representations stored in the image database) one by one. Retrievals may become extremely slow, especially when database search involves time consuming image matching operations. To deal with slow retrieval response times, and high complexity matching, an image database must utilize indexing methods that are faster than sequential scanning methods. In traditional image database systems, the use of indexing to allow database accessing has been well established. Analogously, image indexing techniques have been studied during the last decade to support representation of pictorial information in an image database and also to retrieve information from an image database. The use of significant elements present in images along with their topological relationships as indexes is the basic issue of the indexing methodologies developed to this aim.

BACKGROUND

An image archival and retrieval system is a system in which a large amount of picture data and their related information are stored, retrieved and manipulated. It is indeed necessary to design an image database system that represents images more efficiently. In addition, making the system capable of responding to specified queries to retrieve desired images from the database besides devising robust search techniques to make retrieval process fast and flexible is interesting and more challenging.

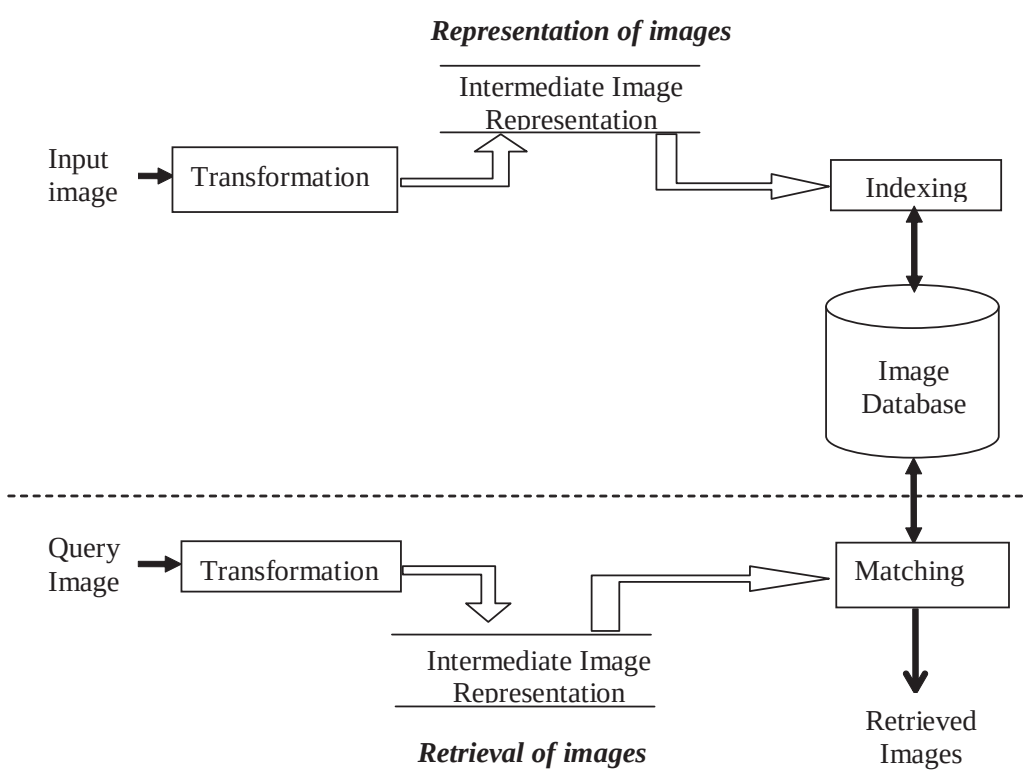
Image Archival and Retrieval System: An Architecture

Crucially, the design of an image database system is a two stage problem. The first stage deals with the representation of images in the image database while the second stage focuses on retrieving images relevant to a given query image as shown in Fig.1 The first stage concentrates on transforming images to intermediate representations through localization and identification of significant elements present in the images and then the mechanism of indexing the intermediate images in the image database. In the second stage, a given query image is subjected to the same transformation process to obtain its intermediate representation which shall subsequently be used in matching.

Image Transformation

Processing of images representing information only at pixel level to have interactive response to high-level user queries is not economically viable if not technologically impossible in a database environment where the number of images tends to be large. Therefore, given an image representing information at pixel level

Figure 1. General architecture of a typical image archival and retrieval system



(lower level), various image processing and understanding techniques are used to identify the domain objects/components and their locations in the image along with their centroid co-ordinates. An image is then obtained by associating a unique name and a meaningful graphic icon with each domain object identified. Intuitively, a domain object is a semantic entity (with respect to an application) contained in an image. For example, in the interior design application, the various furniture and decorative items in an image constitute the domain objects. An image of human face includes eyes, nose, mouth, ears and jaw as the domain objects. As another example, minutiae points with class labels and additional information such as their orientation form domain objects in fingerprint images. Encoding each iconic object present in an image by the respective label produces the corresponding symbolic image/ iconic image/ logical image. Though this task of transforming a physical image into corresponding symbolic image is computationally expensive and difficult, it is performed only once at the time of image insertion into the image database. Moreover this task may be carried out in a semi-automated (human –assisted) scheme or in a fully automated scheme depending upon the domain and complexity of images. However, the transformation of images into corresponding symbolic images, in itself is a challenging research topic.

The storage requirement for representing a symbolic image is relatively negligible when compared to that of representing the image as it is. The decision of arriving at relevant images for a given query is achieved by the usage of symbolic images only. Once the relevant images are identified, only those images are transferred from the central location to a local site.

Image Indexing

To achieve faster retrieval speed and make the retrieval system truly scalable to large size image collection, an effective and efficient indexing is an indispensable part of the whole image database system. The nature of the features extracted from images implies the need for some special purpose storage mechanisms. In order to provide efficient access and retrieval, the data stored in the image database must be organized in some way that enables searching directly based on the extracted features. This ability is often referred to as image indexing. An image database must provide some means of specifying the attributes that should be indexed. Methods should be provided to build these indices automatically. Thus, creation of an image database requires the knowledge of advanced issues in data structures.

Image Database

An image database is a repository of images consisting of a large amount of picture data and their related information. It consists of both symbolic images as well as their corresponding real images. The information stored in an image database can further be referred, retrieved and manipulated.

Image Matching

Once images are stored in an image database, the image database system should provide a means to retrieve images relevant to the specification. Since each image is associated with a feature vector, distance measures that compute distances between these feature vectors are used to find a match between images. Feature vectors usually exist in a very high dimensional space. Due to this high dimensionality, non-parametric approaches, like the nearest neighbor rule, are used for matching purpose. More generally, the matching works on the basis of two well accepted theories i.e., Minimum distance criterion and Maximum likelihood criterion. Minimum distance measure looks for the degree of dissimilarity between the query image and the image being retrieved as a relevant image to the query image, while maximum likelihood works based on the degree of similarity. Since an indexing mechanism based on the features extracted is already framed during representing images in the database, the same can be used to retrieve the images in an efficient way.

Visual Features for Image Retrieval

The area of image retrieval gained its initial attention in the late 70s. Since then, till date, it has attracted many researchers and has become a major focused field of Computer Vision. This is also evident from many special issues of leading journals being dedicated to this topic of research.

A large volume and variety of digital images currently acquired and used in different application domains has given rise to the requirement for intelligent image management and retrieval techniques. In particular,

there is an increased availability of automated image content analysis and description techniques in order to retrieve images efficiently from large collections, based on their visual content such as color, texture and shape information that is present in the images. In the following paragraphs we present a very brief review of the work done on various visual contents. A detailed survey on the approaches proposed and the systems developed in the early years can be found in (Liu et al., 2007; Punitha, 2006).

Color Based

Color not only adds beauty to objects but also gives more information, which can be used as powerful tool in content-based image retrieval. In color based retrieval, given a query image, the goal is to retrieve all the images whose color compositions are similar to the color composition of a given query image. Though the color matching algorithms have their own limitations, such as two semantically different objects possess same color, some researchers are still working in this area to improve the performance of color based retrieval systems (Ozden and Polat, 2007; Sun et al., 2006).

Texture Based

An ability to match on texture similarities can often be more useful in distinguishing between regions of images with same color (such as sky and sea, or leaves and grass). To achieve better performance in measuring texture similarities, a few techniques have been proposed recently (Sastry et al., 2007; Li and Taylor, 2005). Texture queries can be formulated in a similar manner to color queries, by selecting examples of desired textures from a palette, or by supplying an example query image. Similar to color, neither, ability to retrieve images on the basis of only texture similarity does seem very useful.

Shape Based

Shape is a fairly well defined concept compared to color and texture and there is considerable evidence that a natural object is primarily recognized by its shape. The problem of retrieving images containing objects, which are similar to the objects specified in a query image, is transformed into a problem of object recognition, or object classification which is a well known problem in computer vision research. Even recently, a few techniques for image retrieval based on shape were proposed (Chi and Leung, 2007; Alajlan et al., 2006).

Hybridised

A few models, with combined visual features are also explored (Ozden and Polat, 2007; Sun et al., 2006; Hafiane et al., 2006).

Image retrieval systems cited above mainly perform extraction of visual features typically color, shape and texture as a set of uncorrelated characteristics. Such features no doubt provide a global description of images but fail to consider the meanings of portrayed objects and the semantics of images. At a more abstract level of knowledge about the content of images, extraction of object descriptions and their relative positions provide a spatial configuration and a logical representation of images. An image retrieval system should perform similarity matching based on the representation of visual features conveying the content of segmented regions. Besides, it should capture the spatial relationship among the components, in order to face the user expectations. Indeed, we measure the similarity of images solely on the basis of the components' locations, relative to the locations of other known components present in images. For example, a door image may be of any color and with any design and still can be recognized immediately as a door image on the basis of the relative locations of the lock, doorknob and the handle on the door. Based on the spatial scattering of minutiae points, fingerprint image could be better archived and indexed for faster retrieval. Also, functions for retrieval by spatial similarity based on symbolic images are useful in distributed environments where physical images are stored only at a central location while the symbolic images are stored at each local site. Hence, the most appreciated, acclaimed and currently existing way of representing an image is through the perception of spatial relationships existing amongst the components present in it (Lee et al., 2007; Pham and Smeulders, 2006; Bloch et al., 2006, 2005; Punitha and Guru, 2006, 2005(a); Dehak et al., 2005; Castagliola et al., 2005; Maitre, 2005; Guru and Punitha, 2004(a)).

MAJOR FOCUS

Retrieval by spatial similarity has received considerable attention with many applications viz., object recognition (Pham and Smeulders, 2006), GIS (Hafiane et al., 2006), floor planning (Choi et al., 2006). Retrieval by spatial similarity deals with a class of queries that is based on spatial/topological relations (Punitha, 2006) among domain objects. The insinuated methodologies can be broadly classified into two categories, the value oriented which are shown to be insufficient to deal with complicated operations in an intelligent, fast and flexible manner as they work on low level image features, while other alternative object oriented models receive considerable attention. The object oriented methodologies can be further classified as string based approaches, matrix based approaches, hash table based approaches, graph based approaches.

2D string which is the foremost object oriented method brought a significant turn in the field of image retrieval offering many advantages. It used longest common subsequence matching for retrieval of similar images. Although, many string based representations, were inspired by 2D string, the linear string representation given to the spatial relations existing among the components takes nondeterministic-polynomial time complexity during the process of string matching, in addition to being not invariant to image transformations, especially to rotation. In order to reduce the search time and to avoid string matching, hash-oriented methodologies for similarity retrieval based upon the variations of string based approaches were explored. However, hash function-based algorithms require $O(m^2)$ retrieval time in the worst case, where m is the number of iconic objects. Most of the string based and hash based methodologies are not invariant to image transformations. Many matrix based approaches were proposed which tried to satisfy image invariance property to some extent. However, the matrix matching still increased the matching complexity. The graph based algorithms are again not very suitable as similarity matching in graph itself is a research topic.

In order to achieve fast retrieval, indexing search are learnt to be the best. Indexed search is faster than any other ways even for large collections (with more than 100 thousand images) (Lin et al., 2007; Saha et al., 2007; Manolopoulos et al., 2006; Punitha, 2006; Kwok and Leon, 2006; Alajlan et al., 2006; Guru et al., 2003). In addition to effectiveness and efficiency, an important characteristic of an image retrieval system should be that they are invariant to scene conditions, such as changes in viewpoint or image orientation. The formulation of invariant features have been recently addressed by a number of researchers in the field of image databases (Sastry et al., 2007; Shao and Brady, 2006; Punitha and Guru, 2006). To add to these, dynamic databases which support insertion and deletion of images is also gaining interest (Punitha, 2006; Dong and Bhanu, 2003). Most of the currently existing methods also restrict themselves to handle only ones instance of a particular object in the images. Due to its complexity, a very little research, is done in handling multiple instances of objects in images (Punitha and Guru, 2005(b); Guru and Punitha, 2004(b)). Many researchers are also working on identifying/segmenting and labeling objects by the study of low level visual features, which is an essential step for the design of an highly intelligent image archival and retrieval system (Pham and Smeulders, 2006; Shao and Brady, 2006).

FUTURE TRENDS

Despite the fact that semantic web technologies like ontology and semantic search, ostensive search which work on the basis of relevance feedback have received high appreciation, an intelligent IARS, which would retrieve images solely based on the visual features, without human intervention would be always foreseen and will be the most preferred. To assist this goal, efficient and robust techniques for the transformation of physical image into its corresponding logical image will have to be investigated. To achieve this, the possibility of incorporation of image segmentation and labeling techniques must be studied. To make the system intelligent enough to identify objects and also the semantics embedded in images, solely on the visual features, fuzzy theory based approaches could be explored. Moreover, in order to capture the reality being embedded in the physical image, an attempt to make use of unconventional data types viz., interval, range with weightage, multiple etc., in transforming a physical image into a logical image can also be made. Furthermore, discovering altogether a different method to represent logical image obtained through unconventional data analysis is very challenging. Of course, the corresponding retrieval schemes should also be devised. In order to further speed up the task of retrieval and

also to make it best suitable for online/real time applications, the parallel processing techniques may be incorporated. The parallelism can certainly be achieved as the elements present in images could concurrently be located and labeled to obtain logical images.

CONCLUSION

In this chapter, the state-of-art image retrieval specific to spatial reasoning has been highlighted. A very brief literature survey, especially on the recent contributions is made. Some open avenues and few directions are also figured out. In summary, there is a need for an efficient intelligent IARS. The developed IARS is expected to be intelligent enough to represent the images in a realistic manner. The system is intelligent in the sense that it supports knowledge driven image retrieval schemes, not simply based on similarity match but based on image induction too. That is, the algorithms are expected to mine the required knowledge, sometimes if possible, out of the raw data available about images in the image database. This capability would make the system useful in semantic based image retrieval.

REFERENCES

Alajlan N., Kamel M.S., & Freeman G., (2006), Multi-object image retrieval based on shape and topology, Signal Processing: Image Communication, Vol.21, 904–918.

Bloch I., Colliot O., Camara O., & Ge´Raud T., (2005), Fusion of spatial relationships for guiding recognition, example of brain structure recognition in 3D MRI, Pattern Recognition Letters, Vol.26, 449–457.

Bloch I., Colliot O., & Cesar R. M., (2006), On the ternary spatial relation "Between", Systems, Man, and Cybernetics—Part B: Cybernetics, Vol.36(2), 312-327.

Castagliola G., Ferrucci F., & Gravino C., (2005), Adding symbolic information to picture models: definitions and properties, Theoretical Computer Science, Vol.337(1-3), 51-104.

Chi Y., & Leung M.K.H., (2007), ALSBIR: A local-structure-based image retrieval, Pattern Recognition, Vol.40(1), 244-261.

Choi J. W., Kwon D. Y., Hwang J. E., & Lertlakkhanakul J., (2006), Real time management of spatial information of design: A space based floor plan representation of buildings, Automation in Construction, In press, doi:10.1016/j.autcon.2006.08.003.

Dehak S.M.R., Bloch I., & Maître H., (2005), Spatial reasoning with incomplete information on relative positioning, Pattern Analysis and Machine Intelligence, Vol.27(9), 1473 – 1484.

Dong A., & Bhanu B., (2003), Active concept learning for image retrieval in dynamic databases, Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV'03).

Guru D.S., Punitha P., & Nagabhushan P., (2003), Archival and retrieval of symbolic images: an invariant scheme based on triangular spatial relationship, Pattern Recognition Letters, Vol.24(14), 2397-2408.

Guru D.S., & Punitha P., (2004(a)), An invariant scheme for exact match retrieval of symbolic images based upon principal component analysis, Pattern Recognition Letters, Vol.25(1), 73-86.

Guru D.S., & Punitha P., (2004(b)), Similarity retrieval of symbolic images with multiple instances of iconic objects: A novel approach, Proceedings of Fourth Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP), 417-422.

Hafiane A., Chaudhuri S., Seetharaman G., & Zavidovique B., (2006), Region-based CBIR in GIS with local space filling curves to spatial representation, Pattern Recognition Letters, Vol.27(4), 259-267.

Kwok S.H., & Leon Z.J., (2006), Content based object organisation for efficient image retrieval in image databases, Decision Support Systems, Vol.42, 1901-1916.

Lee A.J.T., Hong R.W., Ko W.M., Tsao W.K., & Lin H.H., (2007), Mining spatial association rules in image databases, Information Sciences, Vol.177, 1593 – 1608.

Li S., & Taylor J.S., (2005), Comparison and fusion of multiresolution features for texture classification, Pattern Recognition Letters, Vol.26, 633– 638.

Lin H.Y., Huang P.W., & Hsu K.H., (2007), A new indexing method with high storage utilization and retrieval efficiency for large spatial databases, In-

formation and Software Technology, doi. 10. 1016/ j.infsof.2006.09.005.

Liu Y., Zhang D., Lu G., & Ma W.Y., (2007), A survey on content based image retrieval with high level semantics, Pattern Recognition, Vol.40, 262-282.

Maıˆ Tre H., (2005), Spatial reasoning with incomplete information on relative positioning, Pattern Analysis and Machine Intelligence, Vol.27(9), 1473-1484.

Manolopoulos Y., Nanopoulos A., Papadopoulos A.N., & Theodoridis Y., (2006), R-Trees: theory and applications, Series in Advanced Information and Knowledge Processing, Springer, Berlin.

Ozden M., & Polat E., (2007), A color image segmentation approach for content-based image retrieval, Pattern Recognition, Vol.40, 1318 – 1325.

Pham T.V., & Smeulders A.W.M., (2006), Learning spatial relations in object recognition, Pattern Recognition Letters, Vol.27, 1673–1684.

Punitha P., & Guru D.S., (2005(a)), An invariant scheme for exact match retrieval of symbolic images: triangular spatial relationship based approach, Pattern Recognition Letters, Vol.26(7), 893-907.

Punitha P., & Guru D.S., (2005(b)), A dissimilarity measure based spatial similarity retrieval of symbolic images: a novel approach to handle multiple instances of iconic objects, Society of Statistics, Computer and Applications, Vol.3(1-2), 117-132.

Punitha P., (2006), IARS: Image Archival and Retrieval System (Exploration of spatial similarity based indexing schemes for symbolic image databases), Unpublished Doctoral Dissertation, Department of Studies in Computer Science, University of Mysore, Mysore, India.

Punitha P., & Guru D.S., (2006), An effective and efficient exact match retrieval scheme for image database systems based on spatial reasoning: A logarithmic search time approach, Knowledge and Data Engineering, Vol.18(10), 1368-1381.

Saha S.K., Das A.K., & Chanda B., (2007), Image retrieval based on indexing and relevance feedback, Pattern Recognition Letters, Vol.28, 357–366.

Sastry C.S., Ravindranath M., Pujari A.K., & Deekshatulu B.L., (2007), A modified Gabor function for content based image retrieval, Pattern Recognition Letters, Vol.28, 293–300.

Shao L., & Brady M., (2006), Specific object retrieval based on salient regions, Pattern Recognition, Vol.39, 1932-1948.

Sun J., Zhang X., Cui J., & Zhou J., (2006), Image retrieval based on color distribution entropy, Pattern Recognition Letters, Vol.27, 1122–1126.

KEY TERMS

2D String: An ordered pair *(u, v)*, where *u* and *v* denote a 1-dimensional string containing spatial relations between symbolic objects along the X and Y axes respectively.

Digital Photography: Corresponds to the images captured by digital sensors.

Image Archival and Retrieval System (IARS): A system in which a large amount of picture data and their related information are stored, retrieved and manipulated.

Image Indexing: A way of organizing the data to be stored in the image database in a way that enables searching directly based on the extracted features.

Intelligent IARS: A IARS which performs knowledge driven retrieval.

Symbolic Image: An abstract representation of the physical image with the objects in the image been segmented and labeled.

Texture: Refers to the patterns in an image that has the properties of homogeneity that do not result from the presence of single color or intensity value.

Intelligent Query Answering

Zbigniew W. Ras
University of North Carolina, Charlotte, USA

Agnieszka Dardzinska
Bialystok Technical University, Poland

INTRODUCTION

One way to make Query Answering System (QAS) intelligent is to assume a hierarchical structure of its attributes. Such systems have been investigated by (Cuppens & Demolombe, 1988), (Gal & Minker, 1988), (Gaasterland et al., 1992) and they are called cooperative. Any attribute value listed in a query, submitted to cooperative QAS, is seen as a node of the tree representing that attribute. If QAS retrieves no objects supporting query q, from a queried information system S, then any attribute value listed in q can be generalized and the same the number of objects supporting q in S can increase. In cooperative systems, these generalizations are controlled either by users (Gal & Minker, 1988), or by knowledge discovery techniques (Muslea, 2004).

If QAS for S collaborates and exchanges knowledge with other systems, then it is also called intelligent. In papers (Ras & Dardzinska, 2004, 2006), a guided process of rules extraction and their goal-oriented exchange among systems is proposed. These rules define foreign attribute values for S and they are used to construct new attributes and/or impute null or hidden values of attributes in S. By enlarging the set of attributes from which queries for S can be built and by reducing the incompleteness of S, we not only enlarge the set of queries which QAS can successfully handle but also we increase the overall number of retrieved objects.

So, QAS based on knowledge discovery has two classical scenarios which need to be considered:

- **System is standalone and incomplete.**

Classification rules are extracted and used to predict what values should replace null values before any query is answered.

- **System is distributed with autonomous sites (including site S). User needs to retrieve objects from S satisfying query q containing nonlocal attributes for S.**

We search for definitions of these non-local attributes at remote sites for S and use them to approximate q (Ras & Zytkow, 2000), (Ras & Dardzinska, 2004, 2006).

The goal of this article is to provide foundations and basic results for knowledge-discovery based QAS.

BACKGROUND

Modern query answering systems area of research is related to enhancements of query-answering systems into intelligent systems. The emphasis is on problems in users posing queries and systems producing answers. This becomes more and more relevant as the amount of information available from local or distributed information sources increases. We need systems not only easy to use but also intelligent in handling the users' needs. A query-answering system often replaces human with expertise in the domain of interest, thus it is important, from the user's point of view, to compare the system and the human expert as alternative means for accessing information.

A knowledge system is defined as an information system S coupled with a knowledge base KB which is simplified in (Ras & Zytkow, 2000), (Ras & Dardzinska, 2004, 2006) to a set of rules treated as definitions of attribute values. If information system is distributed with autonomous sites, these rules can be extracted either locally from S (query was submitted to S) or from its remote sites. The initial alphabet of QAS associated with S contains all values of attributes in *S*, called local, and all decision values used in rules from KB. When KB is updated (new rules are added or some deleted), the alphabet for the local query answering

system is automatically changed. It is often assumed that knowledge bases for all sites are initially empty. Collaborative information system (Ras & Dardzinska, 2004, 2006) learns rules describing values of incomplete attributes and attributes classified as foreign for its site called a client. These rules can be extracted at any site but their condition part should use, if possible, only terms which can be processed by the query answering system associated with the client. When the time progresses more and more rules can be added to the local knowledge base which means that some attribute values (decision parts of rules) foreign for the client are also added to its local alphabet. The choice of which site should be contacted first, in search for definitions of foreign attribute values, is mainly based on the number of attribute values common for the client and server sites. The solution to this problem is given in (Ras & Dardzinska, 2006).

MAIN THRUST

The technology dimension will be explored to help clarify the meaning of intelligent query answering based on knowledge discovery and chase.

Intelligent Query Answering for Standalone Information System

QAS for an information system is concerned with identifying all objects in the system satisfying a given description. For example an information system might contain information about students in a class and classify them using four attributes of "hair color", "eye color", "gender" and "size". A simple query might be to find all students with brown hair and blue eyes. When information system is incomplete, students having brown hair and unknown eye color can be handled by either including or excluding them from the answer to the query. In the first case we talk about optimistic approach to query evaluation while in the second case we talk about pessimistic approach. Another option to handle such a query would be to discover rules for eye color in terms of the attributes hair color, gender, and size. These rules could then be applied to students with unknown eye color to generate values that could be used in answering the query. Consider that in our example one of the generated rules said:

(hair, brown) $\wedge$ (size, medium) $\rightarrow$ (eye,brown).

Thus, if one of the students having brown hair and medium size has no value for eye color, then the query answering system should not include this student in the list of students with brown hair and blue eyes. Attributes hair color and size are classification attributes and eye color is the decision attribute.

We are also interested in how to use this strategy to build intelligent QAS for incomplete information systems. If query is submitted to information system S, the first step of QAS is to make S as complete as possible. The approach proposed in (Dardzinska & Ras, 2005) is to use not only functional dependencies to chase S (Atzeni & DeAntonellis, 1992) but also use rules discovered from a complete subsystem of S to do the chasing.

In the first step, intelligent QAS identifies all incomplete attributes used in a query. An attribute is incomplete in S if there is an object in S with incomplete information on this attribute. The values of all incomplete attributes are treated as concepts to be learned (in a form of rules) from S.

Incomplete information in S is replaced by new data provided by Chase algorithm based on these rules. When the process of removing incomplete vales in the local information system is completed, QAS finds the answer to query in a usual way.

Intelligent Query Answering for Distributed Autonomous Information Systems

Semantic inconsistencies are due to different interpretations of attributes and their values among sites (for instance one site can interpret the concept "young" differently than other sites). Different interpretations are also due to the way each site is handling null values. Null value replacement by values suggested either by statistical or knowledge discovery methods is quite common before user query is processed by QAS.

Ontology (Guarino, 1998), (Van Heijst et al., 1997) is a set of terms of a particular information domain and the relationships among them. Currently, there is a great deal of interest in the development of ontologies to facilitate knowledge sharing among information systems.

Ontologies and inter-ontology relationships between them are created by experts in corresponding domain,

but they can also represent a particular point of view of the global information system by describing customized domains. To allow intelligent query processing, it is often assumed that an information system is coupled with some ontology. Inter-ontology relationships can be seen as semantical bridges between ontologies built for each of the autonomous information systems so they can collaborate and understand each other.

In (Ras and Dardzinska, 2004), the notion of optimal rough semantics and the method of its construction have been proposed. Rough semantics can be used to model semantic inconsistencies among sites due to different interpretations of incomplete values of attributes. Distributed chase (Ras and Dardzinska, 2006) is a chase-type algorithm, driven by a client site of a distributed information system DIS, which is similar to chase algorithms based on knowledge discovery and presented in (Dardzinska and Ras, 2005). Distributed chase has one extra feature in comparison to other chase-type algorithms: the dynamic creation of knowledge bases at all sites of DIS involved in the process of solving a query submitted to the client site of DIS.

The knowledge base at the client site may contain rules extracted from the client information system and also rules extracted from information systems at remote sites in DIS. These rules are dynamically updated through the incomplete values replacement process (Ras and Dardzinska, 2004, 2006).

Although the names of attributes are often the same among sites, their semantics and granularity levels may differ from site to site. As the result of these differences, the knowledge bases at the client site and at remote sites have to satisfy certain properties in order to be applicable in a distributed chase.

So, assume that system S = (X,A,V), which is a part of DIS, is queried be user.

Chase algorithm, to be applicable to S, has to be based on rules from the knowledge base D associated with S which satisfies the following conditions:

1. Attribute value used in decision part of a rule from D has the granularity level either equal to or finer than the granularity level of the corresponding attribute in S.
2. The granularity level of any attribute used in the classification part of a rule from D is either equal or softer than the granularity level of the corresponding attribute in S.
3. Attribute used in the decision part of a rule from D either does not belong to A or is incomplete in S.

Assume again that S=(X,A,V) is an information system (Pawlak, 1991), where X is a set of objects, A is a set of attributes (seen as partial functions from X into $2^{(V\times[0,1])}$ and, V is a set of values of attributes from A. By [0,1] we mean the set of real numbers from 0 to 1. Let $L(D)=\{[t \rightarrow v_c] \in D: c \in In(A)\}$ be a set of all rules (called a knowledge-base) extracted initially from the information system S by ERID (Dardzinska and Ras, 2006), where In(A) is a set of incomplete attributes in S.

Assume now that query q(B) is submitted to system S=(X,A,V), where B is the set of all attributes used in q(B) and that $A \cap B \neq \emptyset$. All attributes in $B - [A \cap B]$ are called foreign for S. If S is a part of a distributed information system, definitions of foreign attributes for S can be extracted at its remote sites. Clearly, all semantic inconsistencies and differences in granularity of attribute values among sites have to be resolved first. In (Ras and Dardzinska, 2004) only different granularity of attribute values and different semantics related to different interpretations of incomplete attribute values among sites have been considered.

In (Ras and Dardzinska, 2006), it was shown that query q(B) can be processed at site S by discovering definitions of values of attributes from $B - [A \cap B]$ at the remote sites for S and next use them to answer q(B).

Foreign attributes for S in B, can be also seen as attributes entirely incomplete in S, which means values (either exact or partially incomplete) of such attributes should be ascribed by chase to all objects in S before query q(B) is answered. The question remains, if values discovered by chase are really correct?

Classical approach, to this kind of problems, is to build a simple DIS environment (mainly to avoid difficulties related to different granularity and different semantics of attributes at different sites). As the testing data set we have taken 10,000 tuples randomly selected from a database of some insurance company in Charlotte, NC. This sample table, containing 100 attributes, was randomly partitioned into four subtables of equal size containing 2,500 tuples each. Next, from each of these subtables 40 attributes (columns) have been randomly removed leaving four data tables of the size 2,500×60 each. One of these tables was called a

client and the remaining 3 have been called servers. Now, for all objects at the client site, values of one of the attributes, which was chosen randomly, have been hidden. This attribute is denoted by d. At each server site, if attribute d was listed in its domain schema, descriptions of d using See5 software (data are complete so it was not necessary to use ERID) have been learned. All these descriptions, in the form of rules, have been stored in the knowledge base of the client. Distributed Chase was applied to predict what is the real value of the hidden attribute for each object x at the client site. The threshold value $\lambda = 0.125$ was used to rule out all values predicted by distributed Chase with confidence below that threshold. Almost all hidden values (2476 out of 2500) have been discovered correctly (assuming $\lambda = 0.125$) (Ras & Dardzinska, 2006).

Distributed Chase and Security Problem of Hidden Attributes

Assume now that an information system S=(X,A,V) is a part of DIS and attribute $b \in A$ has to be hidden. For that purpose, we construct S_b=(X,A,V) to replace S, where:

1. $a_S(x) = a_{Sb}(x)$, for any $a \in A-\{b\}$, $x \in X$,
2. $b_{Sb}(x)$ is undefined, for any $x \in X$,
3. $b_S(x) \in V_b$.

Users are allowed to submit queries to S_b and not to S. What about the information system Chase(S_b)? How it differs from S?

If $b_S(x) = b_{Chase(Sb)}(x)$, where $x \in X$, then values of additional attributes for object x have to be hidden in S_b to guarantee that value $b_S(x)$ can not be reconstructed by Chase. Algorithm SCIKD for protection of sensitive data against Chase was proposed in (Im & Ras, 2007).

FUTURE TRENDS

One of the main problems related to semantics of an incomplete information system S is the freedom how new values are constructed to replace incomplete values in S, before any rule extraction process begins. This replacement of incomplete attribute values in some of the slots in S can be done either by chase or/and by a number of available statistical methods. This implies that semantics of queries submitted to S and driven (defined) by query answering system QAS based on chase may often differ. Although rough semantics can be used by QAS to handle this problem, we still have to look for new alternate methods.

Assuming different semantics of attributes among sites in DIS, the use of global ontology or local ontologies built jointly with inter-ontology relationships among them seems to be necessary for solving queries in DIS using knowledge discovery and chase. Still a lot of research has to be done in this area.

CONCLUSION

Assume that the client site in DIS is represented by partially incomplete information system S. When a query is submitted to S, its query answering system QAS will replace S by Chase(S) and next will solve the query using, for instance, the strategy proposed in (Ras and Dardzinska, 2004). Rules used by Chase can be extracted from S or from its remote sites in DIS assuming that all differences in semantics of attributes and differences in granularity levels of attributes are resolved first. We can argue here why the resulting information system obtained by Chase can not be stored aside and reused when a new query is submitted to S? If system S is not frequently updated, we can do that by keeping a copy of Chase(S) and next reusing that copy when a new query is submitted to S. But, the original information system S still has to be kept so when user wants to enter new data to S, they can be stored in the original system. System Chase(S), if stored aside, can not be reused by QAS when the number of updates in the original S exceeds a given threshold value. It means that the new updated information system S has to be chased again before any query is answered by QAS.

REFERENCES

Atzeni, P., DeAntonellis, V. (1992). *Relational Database Theory*, The Benjamin Cummings Publishing Company.

Cuppens, F., Demolombe, R. (1988). Cooperative answering: a methodology to provide intelligent access to databases, *Proceedings of the Second International Conference on Expert Database Systems*, 333-353.

Dardzinska, A., Ras, Z. (2005). CHASE-2: Rule based chase algorithm for information systems of type lambda, *Post-proceedings of the Second International Workshop on Active Mining (AM'2003),* Maebashi City, Japan, LNAI 3430, Springer, 258-270

Dardzinska, A., Ras, Z. (2006). Extracting rules from incomplete decision systems: System ERID, in *Foundations and Novel Approaches in Data Mining*, Studies in Computational Intelligence 9, Springer, 143-154

Gal, A., Minker, J. (1988). Informative and cooperative answers in databases using integrity constraints, *Natural Language Understanding and Logic Programming*, North Holland, 277-300.

Gaasterland, T., Godfrey, P., Minker, J. (1992). Relaxation as a platform for cooperative answering, *Journal of Intelligent Information Systems 1 (3),* 293-321.

Giannotti, F., Manco, G. (2002). Integrating data mining with intelligent query answering,

in *Logics in Artificial Intelligence*, LNCS 2424, 517-520

Guarino, N., ed. (1998). *Formal ontology in information systems*, IOS Press, Amsterdam.

Im, S., Ras, Z. (2007). Protection of sensitive data based on reducts in a distributed knowledge discovery system, *Proceedings of the International Conference on Multimedia and Ubiquitous Engineering (MUE 2007),* in Seoul, South Korea, IEEE Computer Society, 762-766

Muslea, I. (2004). Machine Learning for Online Query Relaxation, *Proceedings of KDD-2004*, in Seattle, Washington, ACM, 246-255

Pawlak, Z. (1991). *Rough sets-theoretical aspects of reasoning about data*, Kluwer.

Ras, Z., Dardzinska, A. (2004). Ontology based distributed autonomous knowledge systems, *Information Systems International Journal 29 (1),* Elsevier, 47-58

Ras, Z., Dardzinska, A. (2006). Solving failing queries through cooperation and collaboration, *World Wide Web Journal 9(2),* Springer, 173-186

Ras, Z., Zytkow, J.M. (2000). Mining for attribute definitions in a distributed two-layered DB system, *Journal of Intelligent Information Systems 14 (2/3),* Kluwer, 115-130

Ras, Z., Zhang, X., Lewis, R. (2007). MIRAI: Multi-hierarchical, FS-tree based music information retrieval system, *Proceedings of RSEISP 2007*, LNAI 4585, Springer, 80-89

Van Heijst, G., Schreiber, A., Wielinga, B. (1997). Using explicit ontologies in KBS development, *International Journal of Human and Computer Studies 46, (2/3),* 183-292.

KEY TERMS

Autonomous Information System: Information system existing as an independent entity.

Chase: Kind of a recursive strategy applied to a database V, based on functional dependencies or rules extracted from V, by which a null value or an incomplete value in V is replaced by a new more complete value.

Distributed Chase: Kind of a recursive strategy applied to a database V, based on functional dependencies or rules extracted both from V and other autonomous databases, by which a null value or an incomplete value in V is replaced by a new more complete value. Any differences in semantics among attributes in the involved databases have to be resolved first.

Intelligent Query Answering: Enhancements of query-answering systems into sort of intelligent systems (capable or being adapted or molded). Such systems should be able to interpret incorrectly posed questions and compose an answer not necessarily reflecting precisely what is directly referred to by the question, but rather reflecting what the intermediary understands to be the intention linked with the question.

Knowledge Base: A collection of rules defined as expressions written in predicate calculus. These rules have a form of associations between conjuncts of values of attributes.

Ontology: An explicit formal specification of how to represent objects, concepts and other entities that are assumed to exist in some area of interest and relationships holding among them. Systems that share the same ontology are able to communicate about domain of discourse without necessarily operating on a globally shared theory. System commits to ontology if its

observable actions are consistent with the definitions in the ontology.

Query Semantics: The meaning of a query with an information system as its domain of interpretation. Application of knowledge discovery and Chase in query evaluation makes semantics operational.

Semantics: The meaning of expressions written in some language, as opposed to their syntax which describes how symbols may be combined independently of their meaning.

On Interacting Features in Subset Selection

Zheng Zhao
Arizona State University, USA

Huan Liu
Arizona State University, USA

INTRODUCTION

The high dimensionality of data poses a challenge to learning tasks such as classification. In the presence of many irrelevant features, classification algorithms tend to overfit training data (Guyon & Elisseeff, 2003). Many features can be removed without performance deterioration, and feature selection is one effective means to remove irrelevant features (Liu & Yu, 2005). Feature selection, also known as variable selection, feature reduction, attribute selection or variable subset selection, is the technique of selecting a subset of relevant features for building robust learning models. Usually a feature is relevant due to two reasons: (1) it is strongly correlated with the target concept; or (2) it forms a feature subset with other features and the subset is strongly correlated with the target concept. Optimal feature selection requires an exponentially large search space ($O(2^n)$, where n is the number of features) (Almual-lim & Dietterich, 1994). Researchers often resort to various approximations to determine relevant features, and in many existing feature selection algorithms, feature relevance is determined by correlation between individual features and the class (Hall, 2000; Yu & Liu, 2003). However, a single feature can be considered irrelevant based on its correlation with the class; but when combined with other features, it can become very relevant. Unintentional removal of these features can result in the loss of useful information and thus may cause poor classification performance, which is studied as attribute interaction in (Jakulin & Bratko, 2003). Therefore, it is desirable to consider the effect of feature interaction in feature selection.

BACKGROUND

The goal of feature selection is to remove irrelevant features and retain relevant ones. We first give the definition of feature relevance as in (John et al., 1994).

Definition 1 (Feature Relevance):

Let F be the full set of features, F_i be a feature and $S_i = F - \{F_i\}$. Let $P(C|S)$ denote the conditional probability of class C given a feature sets. A feature F_i is relevant *iff*

$$\exists S_i' \in S_i, \text{ such that } P(C \mid F_i, S_i') \neq P(C \mid S_i') \qquad (1)$$

Definition 1 suggests that a feature can be relevant, if its removal from a feature set reduces the prediction power of the feature set. A feature, whose removal does not reduce the prediction power of any feature set, is an irrelevant feature and can be removed from the whole feature set without any side-effect. From Definition 1, it can be shown that a feature can be relevant due to two reasons: (1) it is strongly correlated with the target concept; or (2) it forms a feature subset with other features and the subset is strongly correlated with the target concept. If a feature is relevant because of the second reason, there exists feature interaction. Feature interaction is characterized by its irreducibility (Jakulin & Bratko, 2004). We give the definition of *k*th-order below.

Definition 2 (*k*th order Feature Interaction):

Let F be a feature subset with k features $F_1, F_2, ..., F_k$. Let $\Im$ denote a metric that measures the relevance of a feature or a feature subset with the class label. Features $F_1, F_2, ... F_k$ are said to interact with each other *iff*: for an arbitrary partition $S = \{S_1, S_2, S_3, ... , S_l\}$ of F, where $2 \leq l \leq k$ and $S_i \neq \phi$, we have $\forall i \in [1, l]$, $\Im(F) > \Im(S_i)$.

It is clear that identifying either relevant features or *k*th-order feature interaction requires exponential time. Therefore Definitions 1 and 2 cannot be directly applied to identify relevant or interacting features when the dimensionality of a data set is huge. Many efficient feature selection algorithms identify relevant features based on the evaluation of the correlation between the

class and a feature (or a current, selected feature subset). Representative measures used for evaluating feature relevance includes (Liu & Motoda, 1998): distance measures (Kononenko, 1994; Robnik-Sikonja & Kononenko, 2003), information measures (Fleuret, 2004), and consistency measures (Almuallim & Dietterich, 1994), to name a few. Using these measures, feature selection algorithms usually start with an empty set and successively add "good" features to the selected feature subset, the so-called sequential forward selection (SFS) framework. Under this framework, features are deemed relevant mainly based on their individually high correlations with the class, and relevant interacting features of high order may be removed (Hall, 2000; Bell & Wang, 2000), because the irreducible nature of feature interaction cannot be attained by SFS. This motivates the necessity of handling feature interaction in feature selection process.

MAIN FOCUS

Finding high-order feature interaction using Definitions 1 and 2 entails exhaustive search of all feature subsets. Existing approaches often determine feature relevance using the correlation between individual features and the class, thus cannot effectively detect interacting features. Ignoring feature interaction and/or unintentional removal of interacting features might result in the loss of useful information and thus may cause poor classification performance. This problem arouses the research attention to the study of interacting features. There are mainly two directions for handling feature interaction in the process of feature selection: using *information theory* or through *margin maximization*.

Detecting Feature Interaction via Information Theory

Information theory can be used to detect feature interaction. The basic idea is that we can detect feature interaction by measuring the information loss of removing a certain feature. The measure of information loss can be achieved by calculating *interaction information* (McGill, 1954) or McGill's multiple mutual information (Han, 1980). Given three variables, *A*, *B* and *C*, the interaction information of them is defined as:

$$I(A;B;C) = H(AB) + H(BC) + H(AC) - H(A) - H(B) - H(C) - H(ABC)$$
$$= I(A,B;C) - I(A;C) - I(B;C) \quad (2)$$

Here $H(\cdot)$ denotes the entropy of a feature or a feature set. $I(X; Y)$ is the mutual information between X and Y, where X can be a feature set, such as $\{X_1, X_2\}$. Interaction information among features can be understood as the amount of information that is common to all the attributes, but not present in any subset. Like mutual information, interaction information is symmetric, meaning that I(A;B;C) = I(A;C;B) = I(C;B;A) = …. However, interaction information can be negative.

If we set one of the features in the interaction information to be the class, then the interaction information can be used to detect the *2*-way feature interaction as defined in Definition 2. $\Im$, the metric that measures the relevance of a feature or a feature subset with the class label, is defined as the mutual information between a feature or a feature set and the class. Positive interaction information indicates the existence of interaction between features.

Using the interaction information defined in Formula (2), we can only detect *2*-way feature interaction. To detect high order feature interaction, we need to generalize the concept to interactions involving an arbitrary number of attributes. In (Jakulin, 2005) the *k-way interaction information* is defined as:

$$I(S) = -\sum_{T \subseteq S} (-1)^{|S \setminus T|} H(T) \quad (3)$$

Where S is a feature set with k features in it, T is a subset of S and "\" is the set difference operator. $|\cdot|$ measures the cardinality of the input feature set, and $H(T)$ is the entropy for the feature subset T and is defined as:

$$H(T) = -\sum_{v \in T} P(v) \log_2 P(v) \quad (4)$$

According to Definition 3, the bigger the $I(S)$ is, the stronger the interaction between the features in S is. The *k*-way multiple mutual information defined in (Jakulin, 2005) is closely related to the lattice-theoretic derivation of multiple mutual information (Han, 1980), $\Delta h(S) = -I(S)$, and to the set-theoretic derivation of multiple mutual information (Yeung, 1991) and co-information (Bell, 2003) as $I'(S) = (-1)^{|S|} \times I(S)$.

Handling Feature Interaction via Margin Maximization

Margin plays an important role in current research of machine learning. It measures the confidence of a classifier with respect to its predictions. The margin of a classifier can be defined by two ways: sample-margin and hypothesis-margin. Sample-margin, as defined in Support Vector Machine (SVM) (Vapnik 1995), measures the distance between an instance and the decision boundary obtained from the classifier. While, the hypothesis-margin measures the "distance" between two alternative hypotheses (or predictions) which may be derived by a classifier on a certain instance. For classifiers, such as SVM (sample margin) and Adaboost (Freund & Schapire 1997) (hypothesis margin), a large margin ensures high accuracy as well as good generalization capability. Recall the fact that the removal of interacting features results in the loss of useful information and thus cause poor classification performance. It is quite straightforward that we can (explicitly) handle the feature interaction problem by selecting a subset of features that ensure the acquisition of large margin for classifiers. Apparently wrapper model (Koha & John 1997) can be utilized to achieve the margin maximization in feature selection. However, because of the high computation expense and its nature of inheriting bias from the classifier used in the wrapper model, a filter model is usually more preferable for feature selection algorithm design (Blum & Langley 1997).

Mostly, hypothesis margin is used to design selection algorithms in this category. Comparing with sample margin, the advantages of hypothesis margin are: first hypothesis margin is easier for computation, and second, hypothesis margin lower bounds the sample margin. In (Bachrach-et al. 2004), the hypothesis-margin of an instance x with respect to a set of points P is defined as:

$$\theta_P(x) = \frac{1}{2}(\|x - nearmiss(x)\| - \|x - nearhit(x)\|) \quad (5)$$

Here *nearhit(x)* and *nearmiss(x)* denote the nearest point to x in P with the same and different label respectively, and $\|\cdot\|$ is a distance measurement. On an instance x, the hypothesis margin measures the robustness of the prediction of the Nearest Neighbor classifier on the instance. It is clear-cut that a negative or small positive margin means an error or an unstable prediction, and a large positive margin means a correct and stable prediction. Therefore, as the sum of the hypothesis margins on each points, a large positive hypothesis margin on the whole dataset ensures the small error rate as well as the robustness of prediction. Based on this definition, a family of algorithms for feature selection using filter model can be developed (Robnik-Sikonja & Kononenko, 2003; Bachrach-et al. 2004; Navot-et al. 2005). RELIEF (Kira & Rendell, 1992; Robnik-Sikonja & Kononenko, 2003), one of the most successful feature selection algorithms, can be shown to be in this category. Figure 1 shows the RELIEF algorithm.

In Figure 1, w is an N dimension vector whose i-th element corresponding to the i-th feature. T is the total sampling times and N is the number of features. RELIEF tries to update the weight of features according to their contribution of the hypothesis margin on points of S. Therefore measured on the feature subset selected

Figure 1. the RELIEF algorithm

1. initialize the weight vector to zero: $w = 0$
2. for $t = 1 \ldots T$,
   (a) pick randomly an instance x from S
   (b) for $i = 1 \ldots N$,
   $$w_i = w_i + (x_i - nearmiss(x)_i)^2 - (x_i - nearhit(x)_i)^2$$
3. the chosen feature set is $\{i | w_i > \tau\}$ where τ is a threshold

by RELIEF, the hypothesis margin on the dataset S is big. Obviously, in RELIEF, the hypothesis margin maximization is achieved by online optimization (Krumke, 2002).

FUTURE TRENDS

Though feature interaction can be detected by interaction information, detecting high-order feature interactions is still a daunting task, especially when the dimensionality of the data is huge. In the framework of feature selection via margin maximization, feature interaction is handled efficiently, but its detection is implicit. Hence, it is highly desirable to investigate and develop efficient algorithms and new measurements that can effectively detect high-order feature interaction. One possible way is to combine the two frameworks into one, in which a margin-based feature selection algorithm, such as RELIEF, is first used to reduce the original input space to a tractable size, and then using the interaction information to detect feature interaction in the reduced feature space. In real world applications, the detection of interacting features goes beyond accurate classification. For example, in discriminant gene identification, finding interactions between genes involved in common biological functions is a way to get a broader view of how they work cooperatively in a cell. Therefore, it is promising to use feature interaction detection approaches to assist people to acquire better an understanding about the real-world problems in which interacting features exist.

CONCLUSION

Interacting features usually carry important information that is relevant to learning tasks. Unintentional removal of these features can result in the loss of useful information, and eventuate poor classification performance. Detecting feature interaction especially for data with huge dimensionality is computationally expensive. The challenge is to design efficient algorithms to handle feature interaction. In this short article, we present a brief review of the current status and categorize the existing approaches into two groups: *feature interaction detecting via information theory* and *handling feature interaction via margin maximization*. As the demand for finding interacting features intensifies, we anticipate the burgeoning efforts in search of effective and efficient approaches to answer pressing issues arising from many new data-intensive applications.

REFERENCES

Almuallim, H. & Dietterich, T.G. (1994). Learning Boolean Concepts in the Presence of Many Irrelevant Features. Artificial Intelligence, Elsevier, Amsterdam, 69, 279-305.

Bachrach, R.G.; Navot, A. & Tishby, N. (2004). Margin based feature selection - theory and algorithms. Proceeding of International Conference on Machine Learning (ICML), ACM Press.

Bell, A. J. (2003). The co-information lattice. Proceedings of the Fifth International Workshop on Independent Component Analysis and Blind Signal Separation: ICA 2003, edited by S. Amari, A. Cichocki, S. Makino, and N. Murata.

Bell, D.A. & Wang, H. (2000). A formalism for relevance and its application in feature subset selection. Machine Learning, 41, 175-195.

Blum, A.L. & Langley, P. (1997). Selection of Relevant Features and Examples in Machine Learning. Artificial Intelligence, 97, 245-271.

Crammer, K.; Bachrach, R.G.; Navot, A. & Tishby, N. (2002). Margin analysis of the LVQ algorithm. Annual Conference on Neural Information Processing Systems (NIPS).

Fleuret, F.(2004). Fast Binary Feature Selection with Conditional Mutual Information. Journal of Machine Learning Research, MIT Press, 5, 1531-1555.

Freund, Y., and Schapire, R. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. Computer Systems and Science. 55(1):119–139.

Guyon, I. & Elisseeff, A. (2003). An introduction to variable and feature selection. Journal of Machine Learning Research, 3, 1157-1182.

Hall, M.A. (2000). Correlation-based Feature Selection for Discrete and Numeric Class Machine Learning.

Proceeding of International Conference on Machine Learning (ICML), 359-366.

Han, T.S. (1980). Multiple mutual informations and multiple interactions in frequency data. Information and Control, 46(1):26–45.

Jakulin, A. (2005). Machine Learning Based on Attribute Interactions, University of Ljubljana, PhD Dissertation.

Jakulin, A. & Bratko, I. (2003). Analyzing Attribute Dependencies. Proc. of Principles of Knowledge Discovery in Data (PKDD), Springer-Verlag, 229-240

Jakulin, A. & Bratko, I. (2004). Testing the significance of attribute interactions. Proceeding of International Conference on Machine Learning (ICML), ACM Press.

John, G.H.; Kohavi, R. & Pfleger, K. (1994). Irrelevant Feature and the Subset Selection Problem. Proceedings of the Eleventh International Conference on Machine Learning, 121-129.

Kira, K., & Rendell, L. (1992). A practical approach to feature selection. Proceedings of the Ninth International Conference on Machine Learning (ICML), 249–256.

Kohavi, R. & John, G.H. (1997). Wrappers for Feature Subset Selection Artificial Intelligence, 97, 273-324.

Kononenko, I. Bergadano, F. & De Raedt, L. (1994). Estimating Attributes: Analysis and Extension of RELIEF. Proceedings of the European Conference on Machine Learning, Springer-Verlag, 171-182.

Krumke, S.O. (2002). Online Optimization: Competitive Analysis and Beyond. Konrad-Zuse-Zentrum für Informationstechnik Berlin, Technical Report.

Liu, H. & Motoda, H. (1998). Feature Selection for Knowledge Discovery and Data Mining. Boston: Kluwer Academic Publishers.

Liu, H. & Yu, L. (2005). Toward Integrating Feature Selection Algorithms for Classification and Clustering. IEEE Transactions on Knowledge and Data Engineering, 17, 491-502.

McGill, W. J. (1954). Multivariate information transmission. Psychometrika, 19(2):97–116.

Navot, A.; Shpigelman, L.; Tishby, N. & Vaadia, E. (2005). Nearest Neighbor Based Feature Selection for Regression and its Application to Neural Activity. Advances in Neural Information Processing Systems (NIPS).

Robnik-Sikonja, M. & Kononenko, I. (2003). Theoretical and empirical analysis of Relief and ReliefF. Machine Learning, 53, 23-69.

Vapnik, V. (1995). The Nature of Statistical Learning Theory. Springer-Verlag New York, Inc.

Yeung, R. W. (1991). A new outlook on Shannon's information measures. IEEE Trans. On Information Theory, 37:466–474.

Yu, L. & Liu, H. (2003). Feature selection for high-dimensional data: a fast correlation-based filter solution. Proceedings of the twentieth International Conference on Machine Learning (ICML), 856-863.

Zhao, Z. & Liu, H. (2007). Searching for Interacting Features. Proceeding of International Joint Conference on Artificial Intelligence (IJCAI), 1156-1161

KEY TERMS

Feature Selection: An important data preprocessing technique for data mining, helps reduce the number of features, remove irrelevant, redundant, or noisy data, and bring the immediate effects for applications: speeding up a data mining algorithm, improving mining performance such as predictive accuracy and result comprehensibility.

Filter Model: The model relies on general characteristics of the data to evaluate the quality of select features without involving any mining algorithm.

Information Entropy: Information entropy, which is alsocalled Shannon's entropy, is the information-theoretic formulation of entropy. It measures how much information there is in a signal or event. An intuitive understanding of information entropy relates to the amount of uncertainty about an event associated with a given probability distribution. In thermodynamics, entropy is a measurement of the disorder of an isolated system.

Mutual Information: A quantity measures the mutual dependence of variables. It is nonnegative and symmetric.

Margin Maximization: It is a process to find the optimal decision boundary for a classifier, such that the margin (either the hypothesis margin or the sample margin) is maximized.

Support Vector Machines (SVMs): A type of supervised learning methods learns classification boundary through finding optimal hyper-planes between classes.

Wrapper Model: The model requires one predetermined mining algorithm and uses its performance as the evaluation criterion of the quality of selected features. It searches for features better suited to the mining algorithm aiming to improve mining performance

On Interactive Data Mining

Yan Zhao
University of Regina, Canada

Yiyu Yao
University of Regina, Canada

INTRODUCTION

Exploring and extracting knowledge from data is one of the fundamental problems in science. Data mining consists of important tasks, such as description, prediction and explanation of data, and applies computer technologies to nontrivial calculations. Computer systems can maintain precise operations under a heavy information load, and also can maintain steady performance. Without the aid of computer systems, it is very difficult for people to be aware of, to extract, to search and to retrieve knowledge in large and separate datasets, let alone interpreting and evaluating data and information that are constantly changing, and then making recommendations or predictions based on inconsistent and/or incomplete data.

On the other hand, the implementations and applications of computer systems reflect the requests of human users, and are affected by human judgement, preference and evaluation. Computer systems rely on human users to set goals, to select alternatives if an original approach fails, to participate in unanticipated emergencies and novel situations, and to develop innovations in order to preserve safety, avoid expensive failure, or increase product quality (Elm, *et al*., 2004; Hancock & Scallen, 1996; Shneiderman, 1998).

Users possess varied skills, intelligence, cognitive styles, and levels of tolerance of frustration. They come to a problem with diverse preferences, requirements and background knowledge. Given a set of data, users will see it from different angles, in different aspects, and with different views. Considering these differences, a universally applicable theory or method to serve the needs of all users does not exist. This motivates and justifies the co-existence of numerous theories and methods of data mining systems, as well as the exploration of new theories and methods.

According to the above observations, we believe that interactive systems are required for data mining tasks. Generally, interactive data mining is an integration of human factors and artificial intelligence (Maanen, Lindenberg and Neerincx, 2005); an interactive system is an integration of a human user and a computer machine, communicating and exchanging information and knowledge. Through interaction and communication, computers and users can share the tasks involved in order to achieve a good balance of automation and human control. Computers are used to retrieve and keep track of large volumes of data, and to carry out complex mathematical or logical operations. Users can then avoid routine, tedious and error-prone tasks, concentrate on critical decision making and planning, and cope with unexpected situations (Elm, *et al*., 2004; Shneiderman, 1998). Moreover, interactive data mining can encourage users' learning, improve insight and understanding of the problem to be solved, and stimulate users to explore creative possibilities. Users' feedback can be used to improve the system. The interaction is mutually beneficial, and imposes new coordination demands on both sides.

BACKGROUND

The importance of human-machine interaction has been well recognized and studied in many disciplines. One example of interactive systems is an information retrieval system or a search engine. A search engine connects users to Web resources. It navigates searches, stores and indexes resources and responses to users' particular queries, and ranks and provides the most relevant results to each query. Most of the time, a user initiates the interaction with a query. Frequently, feedback will arouse the user's particular interest, causing the user to refine the query, and then change or adjust further interaction. Without this mutual connection, it would be hard, if not impossible, for the user to access these resources, no matter how important and how relevant

they are. The search engine, as an interactive system, uses the combined power of the user and the resources, to ultimately generate a new kind of power.

Though human-machine interaction has been emphasized for a variety of disciplines, until recently it has not received enough attention in the domain of data mining (Ankerst, 2001; Brachmann & Anand, 1996; Zhao & Yao, 2005). In particular, the human role in the data mining processes has not received its due attention. Here, we identify two general problems in many of the existing data mining systems:

1. Overemphasizing the automation and efficiency of the system, while neglecting the adaptiveness and effectiveness of the system. Effectiveness includes human subjective understanding, interpretation and evaluation.
2. A lack of explanations and interpretations of the discovered knowledge. Human-machine interaction is always essential for constructing explanations and interpretations.

To study and implement an interactive data mining system, we need to pay more attention to the connection between human users and computers. For cognitive science, Wang and Liu (2003) suggest a relational metaphor, which assumes that relations and connections of neurons represent information and knowledge in the human brain, rather than the neurons alone. Berners-Lee (1999) explicitly states that "in an extreme view, the world can be seen as only connections, nothing else." Based on this statement, the World Wide Web was designed and implemented. Following the same way of thinking, we believe that interactive data mining is sensitive to the capacities and needs of both humans and machines. A critical issue is not how intelligent a user is, or how efficient an algorithm is, but how well these two parts can be connected and communicated, adapted, stimulated and improved.

MAIN THRUST

The design of interactive data mining systems is highlighted by the process, forms and complexity issues of interaction.

Processes of Interactive Data Mining

The entire knowledge discovery process includes data preparation, data selection and reduction, data pre-processing and transformation, pattern discovery, pattern explanation and evaluation, and pattern presentation (Brachmann & Anand, 1996; Fayyad, *et al*., 1996; Mannila, 1997; Yao, Zhao & Maguire, 2003; Yao, Zhong & Zhao, 2004). In an interactive system, these phases can be carried out as follows:

- Interactive data preparation observes raw data with a specific format. Data distribution and relationships between attributes can be easily observed.
- Interactive data selection and reduction involves the reduction of the number of attributes and/or the number of records. A user can specify the attributes of interest and/or data area, and remove data that is outside of the area of concern.
- Interactive data pre-processing and transformation determines the number of intervals, as well as cut-points for continuous datasets, and transforms the dataset into a workable dataset.
- Interactive pattern discovery interactively discovers patterns under the user's guidance, selection, monitoring and supervision. Interactive controls include decisions made on search strategies, directions, heuristics, and the handling of abnormal situations.
- Interactive pattern explanation and evaluation explains and evaluates the discovered pattern if the user requires it. The effectiveness and usefulness of this are subject to the user's judgement.
- Interactive pattern presentation visualizes the patterns that are perceived during the pattern discovery phase, and/or the pattern explanation and evaluation phase.

Practice has shown that the process is virtually a loop, which is iterated until satisfying results are obtained. Most of the existing interactive data mining systems add visual functionalities into some phases, which enable users to invigilate the mining process at various stages, such as raw data visualization and/or final results visualization (Brachmann & Anand, 1996; Elm, *et al*., 2004). Graphical visualization makes it easy to identify and distinguish the trend and distribution. This is a necessary feature for human-machine interaction,

but is not sufficient on its own. To implement a good interactive data mining system, we need to study the types of interactions users expect, and the roles and responsibilities a computer system should take.

Forms of Interaction

Users expect different kinds of human-computer interactions: proposition, information/guidance acquisition, and manipulation. These interactions proceed with the entire data mining process we mentioned above to arrive at desirable mining results.

Users should be allowed to make propositions, describe decisions and selections based on their preference and judgement. For example, a user can state an interested class value for classification tasks, express a target knowledge representation, indicate a question, infer features for explanation, describe a preference order of attributes, set up the constraints, and so on. Subjects of propositions differ among the varying views of individuals. One may initiate different propositions at different times based on different considerations at different cognitive levels. The potential value consideration enters in to the choice of proposition.

Information acquisition is a basic form of interaction associated with information analysis. Information might be presented in various fashions and structures. Raw data is raw information. Mined rules are extracted knowledge. Numerous measurements show the information of an object from different aspects. Each data mining phase contains and generates much information. An object might be changed; the information it holds might be erased, updated or manipulated by the user in question. Benchmarks, official standards and de facto standards are valuable reference knowledge, which can make it easier to learn and evaluate new applications. In general, information acquisition can be conducted by granular computing and hierarchy theory. A granule in a higher level can be decomposed into many granules in a lower level, and conversely, some granules in a lower level can be combined into a granule in a higher level. A granule in a lower level provides a more detailed description than that of a parent granule in the higher level, and a granule in a higher level has a more abstract description than a child granule in the lower level. Users need to retrieve the information in an interactive manner, namely, "show it correctly when I want to or need to see it, and in an understandable format."

Guidance acquisition is another form of interaction. A consultant role that an interactive system can play is to provide knowledge or skills that the user does not have in-house, for example, doing an evaluation or providing an analysis of the implications of environmental trends. To achieve this expert role, the interactive system must be able to "understand" the human proposition, and be able to make corresponding inferences. Guidance is especially useful while the domain is complex and the search space is huge. To achieve guidance, the system needs to store an extra rule base (usually serving as a standard or a reference), and be context aware. The inference function helps users to pay attention to items that are easily ignored, considered as "boundary" issues, or are important but not part of the current focus. The inference function takes the role and responsibility of a consultant. It ensures the process develops in a more balanced manner.

Manipulation is the form of interaction that includes selecting, retrieving, combining and changing objects, using operated objects to obtain new objects. Different data mining phases require different kinds of manipulations. Interactive manipulations obligate the computer system to provide necessary cognitive supports, such as: a systematic approach that uses an exhaustive search or a well-established, recursive search for solving a problem in a finite number of steps; a heuristic approach that selectively searches a portion of a solution space, a sub-problem of the whole problem, or a plausible solution according to the user's special needs; and an analogy approach that uses known solutions to solve an existing problem (Chiew & Wang, 2004; Matlin, 1998; Mayer, 1992; Ormrod, 1999). In addition, interactive systems should allow users to build their own mental buildings using the standard blocks. The blocks can be connected by functions similar to the pipe command in UNIX systems. What this means is that the standard output of the command to the left of the pipe is sent as standard input of the command to the right of the pipe. A result of this interaction is that users can define their own heuristics and algorithms.

The interaction should be directed to construct a reasonable and meaningful cognitive structure to each user. To a novice, the constructive operation is the psychological paradigm in which one constructs his/her own mental model of a given domain; to an expert, the constructive operation is an experienced practice containing anticipation, estimation, understanding and management of the domain.

Figure 1 illustrates the process and the forms of interactive data mining. A particular interactive data mining system can involve interactions of all four forms at six different phases.

Complexity of Interactive Data Mining Systems

Because of the special forms of interaction, complexity issues often raise concerns during implementation. Weir (1991) identified three sources of complexity in interactive applications.

Complexity of the domain: The domain can be very complex because of the size and type of data, the high dimensionality and high degree of linkage that exist in the data. Modelling the domain to a particular search space is essential. Some search spaces may embody a larger number of possible states than others. Knowledge may be not determined by a few discrete factors but by a compound of interrelated factors.

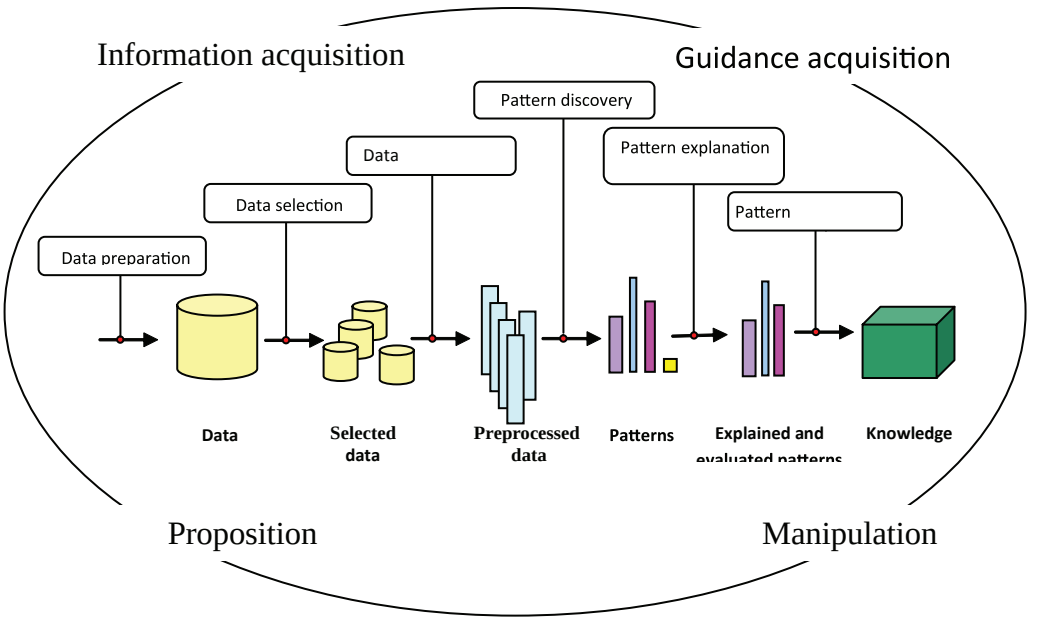
Complexity of control: The complexity of a specific control studies how much time and memory space a chosen computer routine/algorithm may take. It is characterized by its search direction, heuristic, constraint and threshold. Different routines/algorithms have different complexities of control. Normally, a complex domain yields a complex search space, and requires a complex control for searching solutions in the search space.

Complexity of interaction: Complexity of interaction concerns the execution issues of the four interaction forms, some of which are: deciding the degree of involvement of a specific form, scheduling process, doing, undoing, iteration and rollback of a specific control, goal setting and resetting, visualization and recommendation. The greater user demand is, the more complex the overall system becomes.

Implementation Examples

We have implemented an interactive classification system using a granule network (Zhao & Yao, 2005). A granule network systematically organizes all the subsets of the universe and formulas that define the subsets. A consistent classification task can be understood as a search for the distribution of classes in a granule network defined by the descriptive attribute set. Users can freely decide to use a partition-based method, a covering-based method, or a hybrid method for facilitating the search. Classification information can be easily retrieved in the form of a tree-view, a pie chart, a bar chart and/or a pivot table representation. The measurements of attribute and attribute-values are listed. These help the user to judge and select one for

Figure 1. Interactive data mining



splitting. Measures can be chosen from the pre-defined measurement set, or can be composed by the user. Users can validate the mined classification rules at any given time, continue or cease the training process according to the evaluation, split the tree node for higher accuracy, or remove one entire tree branch for simplicity.

Another implementation for interactive attribute selection is currently under construction. In order to keep the original interdependency and distribution of the attribute, the concept of reduct in rough set theory is introduced (Pawlak, 1991). Therefore, the selected attribute set is individually necessary and jointly sufficient for retaining all the information contained in the original attribute set. In this system, users can state a preference order of attributes, satisfying a weak order. Based on this order, a reduct that is most consistent, instead of a random reduct among many, can be computed and presented. Different construction strategies, such as add, add-delete and delete approaches, can be selected. Users can set their preferred attribute order once, or change the order dynamically in order to evaluate different results. In this case, users are allowed to choose a target reduct that is able to preserve accuracy, cost and utility, or distribution property. When a certain reduct is too complicated or too expensive to obtain, an approximate reduct can be constructed.

An interactive explanation-oriented system is our third implementation. The subjects selected for explanation, the explanation context, the explanation construction methods, as well as the explanation evaluation methods all highly dependent upon the preference of an individual user. Please refer to another paper (Yao, Zhao & Maguire, 2003) for further details on this topic.

FUTURE TRENDS

Interactive analysis and mining combines the power of both human users and computer systems. It relies on powerful intuition, analytical skills, insight, and creativity of humans, and fast processing speed, huge storage, and massive computational power of computers. Prototype systems will be implemented to demonstrate the usefulness of the proposed theoretical framework. The seamless integration of humans and computer systems may require the development of multilevel interactive systems, i.e., interaction applied from a low level to a high level, or from fully manual to fully automatic.

From the application point of view, interactive data analysis and mining plays a supporting role for a user. This enables us to design and implement next generation systems that support effective usage of data, for example, decision support systems, business support systems, research support systems and teaching support systems. Considerable research remains to be done.

CONCLUSION

The huge volume of raw data is far beyond a user's processing capacity. One goal of data analysis and mining is to discover, summarize and present information and knowledge from data in concise and human-understandable forms. It should be realized that, at least in the near future, insight about data, as well as its semantics, may not be achieved by a computer system alone. Users, in fact, need to interact with and utilize computer systems as research tools to browse, explore and understand data, and to search for knowledge and insight from data.

Implementing interactive computer systems is an emerging trend in the field of data mining. It aims to have human involvement in the entire data mining process in order to achieve an effective result. This interaction requires adaptive, autonomous systems and adaptive, active users. The performance of these interactions depends upon the complexities of the domain, control, and the available interactive approaches.

REFERENCES

Ankerst, M. (2001) Human involvement and interactivity of the next generations' data mining tools, *ACM SIGMOD Workshop on Research Issues in Data mining and Knowledge Discovery*, Santa Barbara, CA.

Berners-Lee, T. (1999) *Weaving the Web - The Original Design and Ultimate Destiny of the World Wide Web by its Inventor*, Harper Collins Inc.

Brachmann, R. & Anand, T. (1996) The process of knowledge discovery in databases: a human-centered approach, *Advances in Knowledge Discovery and Data mining*, AAAI Press & MIT Press, Menlo Park, CA, 37-57.

Chiew, V. & Wang, Y. (2004) Formal description of the cognitive process of problem solving, *Proceedings of International Conference of Cognitive Informatics*, 74-83.

Elm, W.C., Cook, M.J., Greitzer, F.L., Hoffman, R.R., Moon, B. & Hutchins, S.G. (2004) Designing support for intelligence analysis, *Proceedings of the Human Factors and Ergonomics Society*, 20-24.

Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P. & Uthurusamy, R. (Eds.) (1996) *Advances in Knowledge Discovery and Data mining*, AAAI/MIT Press.

Hancock, P.A. and Scallen, S.F. (1996) The future of function allocation, *Ergonomics in Design*, 4(4), 24-29.

Maanen, P., Lindenberg, J. and Neerincx, M.A. (2005) Integrating human factors and artificial intelligence in the development of human-machine cooperation, *Proceedings of International Conference on Artificial Intelligence*, 10-16.

Mannila, H. (1997) Methods and problems in data mining, *Proceedings of International Conference on Database Theory*, 41-55.

Matlin, M.V. (1998) *Cognition*, fourth edition, Harcount Brace Company.

Mayer, R.E. (1992) *Thinking, Problem Solving, Cognition*, second edition, W.H. Freeman Company.

Ormrod, J.E. (1999) *Human Learning*, third edition, Prentice-Hall, Inc., Simon and Schuster/A Viacom Company.

Pawlak, Z. (1991) *Rough Sets: Theoretical Aspects of Reasoning about Data*, Kluwer Academic Publishers, Dordrecht.

Shneiderman, B. (1998) *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, third edition, Addison-Wesley.

Wang, Y.X. & Liu, D. (2003) On information and knowledge representation in the brain, *Proceedings of International Conference of Cognitive Informatics*, 26-29.

Weir, G.R. (1991) Living with complex interactive systems, in: Weir, G.R. and Alty, J.L. (Eds.) *Human-Computer Interaction and Complex Systems*, Academic Press Ltd.

Yao, Y.Y., Zhao, Y. & Maguire, R.B. (2003) Explanation-oriented association mining using rough set theory, *Proceedings of Rough Sets, Fuzzy Sets and Granular Computing*, 165-172.

Yao, Y.Y., Zhong, N. & Zhao, Y. (2004) A three-layered conceptual framework of data mining, *Proceedings of ICDM Workshop of Foundation of Data mining*, 215-221.

Zhao, Y. & Yao, Y.Y. (2005) Interactive user-driven classification using a granule network, *Proceedings of International Conference of Cognitive Informatics*, 250-259.

Zhao, Y. & Yao, Y.Y. (2005) On interactive data mining, *Proceedings of Indian International Conference on Artificial Intelligence*, 2444-2454.

KEY TERMS

Complexity of Interactive Data Mining: Complexity of the domain, complexity of control and complexity of interaction. The greater user demand, the more complex the overall system becomes.

Forms of Interactive Data Mining: Proposition, information and guidance acquisition, and manipulation.

Interactive Data Mining: An integration of human factors and artificial intelligence. An interactive system thus is an integration of a human user with a computer machine. The study of interactive data mining and interactive systems is directly related to cognitive science.

Process of Interactive Data Mining: Interactive data preparation, interactive data selection and reduction, interactive data pre-processing and transformation, interactive pattern discovery, interactive pattern explanation and evaluation, and interactive pattern presentation.

Interest Pixel Mining

Qi Li
Western Kentucky University, USA

Jieping Ye
Arizona State University, USA

Chandra Kambhamettu
University of Delaware, USA

INTRODUCTION

Visual media data such as an image is the raw data representation for many important applications, such as image retrieval (Mikolajczyk & Schmid 2001), video classification (Lin & Hauptmann, 2002), facial expression recognition (Wang & Ahuja 2003), face recognition (Zhao, Chellappa, Phillips & Rosenfeld 2003), etc. Reducing the dimensionality of raw visual media data is highly desirable since high dimensionality may severely degrade the effectiveness and the efficiency of retrieval algorithms. To obtain low-dimensional representation of visual media data, we can start by selecting good low-level features, such as colors, textures, and interest pixels (Swain & Ballard 1991; Gevers & Smeulders 1998; Schmid, Mohr & Bauckhage 2000).

Pixels of an image may hold different interest strengths according to a specific filtering or convolution technique. The pixels of high interest strengths are expected to be more repeatable and stable than the pixels of low interest strengths across various imaging conditions, such as rotations, lighting conditions, and scaling. Interest pixel mining aims to detect a set of pixels that have the best repeatability across imaging conditions. (An algorithm for interest pixel mining is called a *detector*.) Interest pixel mining can be formulated into two steps: i) interest strength assignment via a specific filtering technique; and ii) candidate selection. The second step, candidate selection, plays an important role in preventing the output of interest pixels from being jammed in a small number of image regions in order to achieve best repeatability.

Based on interest pixels, various image representations can be derived. A straightforward scheme is to represent an image as a collection of local appearances—the intensities of neighboring pixels—of interest pixels (Schmid & Mohr 1997). By ignoring the spatial relationship of interest pixels, this "unstructured" representation requires no image alignment, i.e., free from establishing pixel-to-pixel correspondence among imaging objects by image transformations such as rotation, translation, and scaling. Furthermore, the unstructured representation is very robust with respect to outlier regions in a retrieval application. However, the retrieval cost under unstructured representation is extremely expensive. In the context of face recognition, feature distribution is introduced to capture both global and local information of faces (Li, Ye & Kambhamettu 2006A). A limitation of feature distribution is the assumption of image alignment. A promising trend on interest pixel based representation is to build graph or tree representation for each image and measure the similarity of two images by the edit distance of their graphs or trees (Zhang & Shasha 1989). But as we will see in the later section, this trend is strongly supported by a recently proposed interest pixel mining method (Li, Ye & Kambhamettu 2008).

BACKGROUND

Most previous studies on interest pixel mining focus on exploring filtering techniques, i.e., the first step of an interest pixel detector, which leads to several widely-used filtering techniques such as gradient autocorrelation (Harris & Stephens 1988), and Difference of Gaussian (Lowe 2004). For the second step, previous studies usually adopt the so-called non-maximum suppression as the candidate selection scheme (Harris & Stephens 1988; Schmid, Mohr & Bauckhage 2000; Lowe 2004). Assume each image pixel has been assigned an interest strength. Non-maximum suppression

resets the strength of a pixel to zero, i.e., eliminates its candidacy, if it is not a local maximum. Non-maximum suppression is very effective in preventing interest pixels from being jammed in a small number of image regions. However, studies also show that non-maximum suppression over-suppresses good candidates of interest pixels if an image is weakly-textured, e.g., a face image. Besides the issue of over-suppression, non-maximum suppression may also destroy local geometry information (Li, Ye & Kambhamettu 2008). Brown et. al. proposed adaptive non-maximum suppression to obtain spatially well distributed interest pixels over images (Brown Szeliski & Winder 2005). Instead of using a fixed-size suppression window, adaptive non-maximum suppression dynamically decreases the size of the suppression window. Adaptive non-maximum suppression has been shown to be competitive to the standard one in image mosaicing.

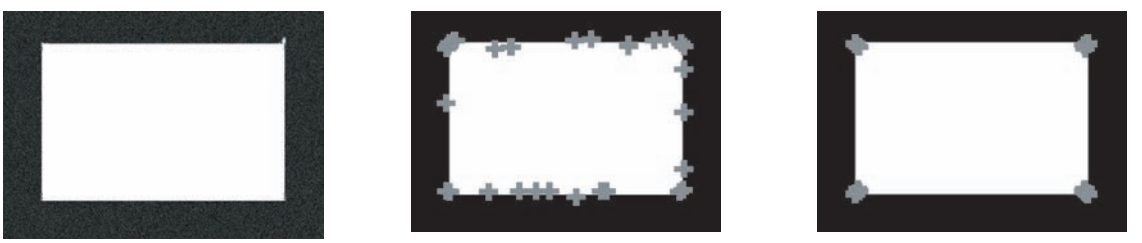
MAIN FOCUS

Li, Ye & Kambhamettu (2008) proposed a novel candidate selection scheme, called imbalance oriented selection that chooses image pixels whose zero-/first-order intensities are clustered into two imbalanced classes (in size), for interest pixel mining. The basic motivation for imbalance oriented selection is to minimize the occurrences of edge pixels. (An edge pixel is a pixel on the boundary of an object or a scene.) It is worth noting that edge pixels are usually not good features in the context of image retrieval. This is because they have similar local appearances, which increases uncertainty in matching local appearances. Besides the issue of ambiguity, the existence of large numbers of edge pixels can also result in high computational cost in high-level applications. Figure 1 shows the key difference between non-maximum suppression and imbalance oriented selection, with a simple image: a white board with a black background. In this circumstance, the edge pixels on the four boundaries of the white board are expected to have distinctly larger strength than other pixels. Furthermore, due to the reality of the existence of noise, the strength of an edge pixel may be slightly different from the strength of its neighboring edge pixels. (The image of a white board in Figure 1 contains 0.1% noise.) So, after non-maximum suppression, the output of interest pixels consists of some edge pixels scattered around the four boundaries. However, with an imbalance oriented scheme, the output of interest pixels only consists of a few image pixels around the four corners of the white board, due to the stronger ability of the scheme in suppressing edge pixels.

FEATURE DISTRIBUTION

A feature distribution of an image represents the number of occurrences of interest pixels in regular grids of an image plane. Figure 2 shows the flow of generating a feature distribution for a simple input image. Feature distribution has been shown a good representation

Figure 1. 30 interest pixels in the image of a white board with a black background

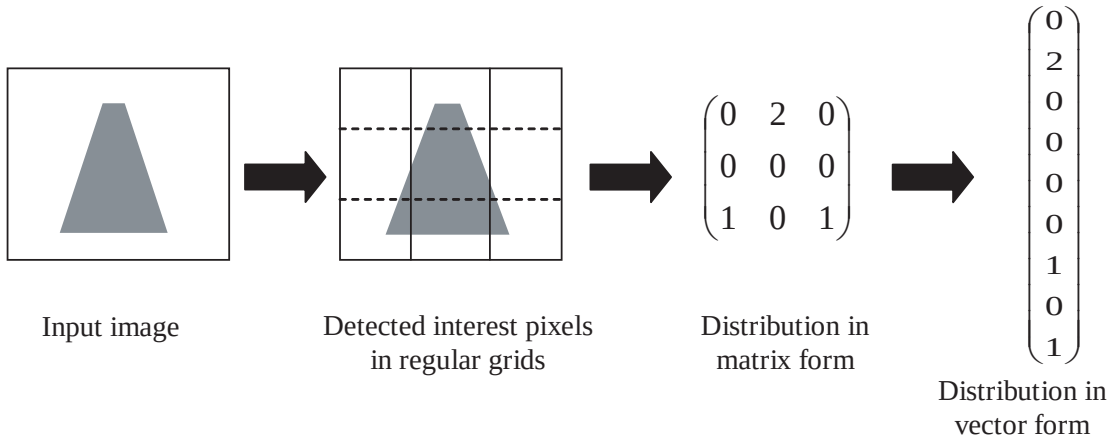


Input image

Non-maximum suppression

Imbalance oriented selection

Figure 2. The flow of generating a feature distribution



technique for face recognition and facial expression recognition (Li, Ye & Kambhamettu 2006A).

It has also been shown that imbalance oriented selection has stronger ability in preserving local geometry structures than non-maximum suppression and achieves better performance in various recognition tasks. A limitation of feature distribution based image representation is the assumption of image alignment. Automatic alignment of a general object is far from straightforward in many scenarios.

UNSTRUCTURED REPRESENTATION

A scheme to overcome the alignment sensitivity issue of feature distribution is to give up the global structure information, and use the local information only. An image is now represented as a set of local appearances of interest pixels (Schmid & Mohr 1997). A database of images is a collection of local appearances of interest pixels of all images along with a label per local appearance. To retrieve a query image from an image database is basically to inquire a set of local appearances of the query image. With the unstructured representation (no preservation of spatial relationship information among interest pixels), measuring the similarity of two images becomes robust with respect to outlier regions, in addition to the advantage of no alignment requirement. A limitation of the unstructured representation is the expensive retrieval cost. Addressing the application of face recognition, Li, Ye & Kambhamettu (2006B) proposed an adaptive framework to integrate the advantages of the structured and unstructured representation.

Specifically, the framework first determines whether an inquiry face image can be successfully aligned via automatic localization of the centers of two eyes. If the alignment can be achieved successfully, the inquiry face image is then represented as a single vector of the entire facial appearance that is inquired in a database of the structured representation of face images of known identities. Otherwise, the inquiry face image is represented as a set of vectors, each of which is associated with a local facial appearance and is inquired in a database of the unstructured representation of face images of known identities.

FUTURE TRENDS

To relax the requirement of image alignment, we may consider graph and furthermore tree representation. For example, based on a set of interest pixels, we can first build up a directed graph, where a node is associated with an interest pixel. Then we can design a certain

scheme to assign links among nodes by the global appearance information. Furthermore, we may obtain a component graph where a component consists of a subset of interest pixels in the same "object''. The above effort will lead us to a segmentation of an image. Figure 3 presents two toy images: a human face and a car, each of which consists of a hierarchy of objects. For example, a face has two eyes and each eye in turn contains one eyeball. Thus, we may use the enclosure/surrounding relationship among objects (components) to build a tree for an input image. The similarity of two images can then be computed by the tree edit distance, i.e., the number of operations of insertion, deletion or re-labeling in order to convert one tree to the other tree (Zhang & Shasha 1989). The idea of tree representation via interest pixels appears to be simple, it is however very difficult to succeed if interest pixels are mined via non-maximum suppression because of its two limitations. First, those interest pixels may include many edge pixels (as shown in Figure 1), which will result in much more noisy component graphs than interest pixels mined via imbalance oriented selection. Second, non-maximum suppression does not have mechanism to distinguish inner or outer pixel regarding a particular boundary, and so it is not reliable to use those interest pixels for segmentation and the enclosure relationship among components. So, the introduction of imbalance oriented selection becomes significant for this future trend.

CONCLUSION

Interest pixels are effective low-level features for image/video representation. Most existing methods for interest pixel mining apply non-maximum suppression to select good candidates. However non-maximum suppression over-suppresses good candidates of interest pixels if an image is weakly-textured and may also destroy local geometry information. Imbalance oriented selection was introduced to overcome the limitation of non-maximum suppression. Three different image representations via interest pixels are described along with the comparison of their advantage and disadvantage.

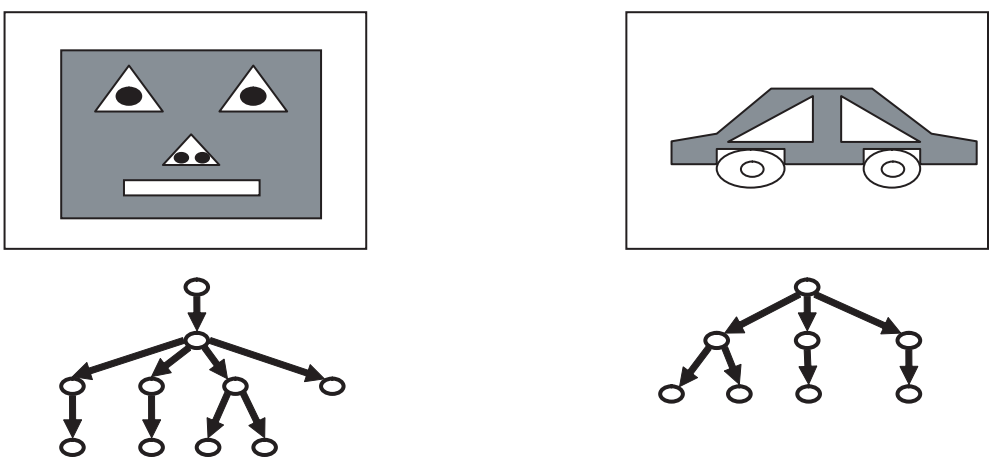
REFERENCES

Brown M., Szeliski R., & Winder S. (2005). Multi-image matching using multi-scale oriented patches. In IEEE Conference on Computer Vision and Pattern Recognition, volume 1, pages 510–517.

Harris C., & Stephens M. (1988). A combined corner and edge detector. In Proc. 4th Alvey Vision Conference, Manchester, pages 147-151.

Gevers T., & Smeulders A.W.M. (1998). Image indexing using composite color and shape Invariant

Figure 3. Tree based image representation (each node represents an image component)



features. In International Conference on Computer Vision, pages 576-581.

Li Q., Ye J., & Kambhamettu C. (2006A). Spatial Interest Pixels (SIPs): Useful Low-Level Features of Visual Media Data. Multimedia Tools and Applications, 30 (1): 89–108.

Li Q., Ye J., & Kambhamettu C. (2006B). Adaptive Appearance Based Face Recognition. In International Conference on Tools with Artificial Intelligence, pages 677-684.

Li Q., Ye J., & Kambhamettu C. (2008). Interest Point Detection Using Imbalance Oriented Selection. Pattern Recognition, Vol. 41, No. 2, Pages 672-688, 2008.

Lin W-H, & Hauptmann A. (2002) News video classification using SVM-based multimodal classifiers and combination strategies. In ACM Multimedia, Juan-les-Pins, France, pages 323–326.

Lowe D.G. (2004) Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 60(2):91–110.

Mikolajczyk K. & Schmid C. (2001). Indexing based on scale invariant interest points. In IEEE International Conference on Computer Vision, volume I, pages 525-531, Vancouver, Canada.

Olague G. & Hernandez B. (2005). A new accurate and flexible model-based multi-corner detector for measurement and recognition. Pattern Recognition Letters, 26(1):27-41.

Perona P. & Moreels P. (2007). Evaluation of features detectors and descriptors based on 3D objects. International Journal on Computer Vision, 73(3): 263-284.

Reisfeld D., Wolfson H., & Yeshurun Y. (1990). Detection of interest points using symmetry. In IEEE International Conference on Computer Vision, pages 62-65.

Schmid C., Mohr R., & Bauckhage C. (2000) Evaluation of interest point detectors. International Journal on Computer Vision, 37(2):151–172.

Schmid C. & Mohr R. (1997) Local grayvalue invariants for image retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(5): 530-535.

Smith S. & Brady J. (1997). SUSAN-a new approach to low level image processing. International Journal of Computer Vision, 23(1):45-78.

Swain M. & Ballard D. (1991). Color indexing. International Journal on Computer Vision 7:11-32.

Trujillo L. & Olague G. (2006). Using evolution to learn how to perform interest point detection. In International Conference on Pattern Recognition. August 20-24, 2006. Hong Kong, China, pages 211-214.

Wang H. & Brady M. (1995). Real-time corner detection algorithm for motion estimation. Image and Vision Computing, 13(9):695-703.

Wang H. & Ahuja N. (2003). Facial expression decomposition. In International Conference on Computer Vision, pages 958-965.

Yacoob Y. & Davis L. (2002). Smiling faces are better for face recognition. In IEEE International Conference on Automatic Face and Gesture Recognition, pages 59-64.

Zhang K. & Shasha D. (1989). Simple fast algorithms for the editing distance between trees and related problems. SIAM J. Computing, 18(6):1245-1252.

Zhao W., Chellappa R., Phillips P., & Rosenfeld A. (2003). Face recognition in still and video images: A literature survey. ACM Computing Surveys, 35(4):399-458.

KEY TERMS

Detector: An algorithm that finds a number of image pixels of strongest interest strength.

Edge Pixel: A pixel on the boundary of an object or a scene. An edge pixel is also called a boundary pixel.

Feature Distribution: A representation of the number of occurrences of interest pixels in regular grids of an image plane.

Image Alignment: A procedure of establishing pixel-to-pixel correspondence among imaging objects by transformations such as rotation, translation, and scaling.

Imbalance Oriented Selection: A candidate selection scheme that chooses image pixels whose zero-/first-order intensities are clustered into two imbalanced classes (in size), as candidates.

Interest Pixel: A pixel that has stronger interest strength than other pixels in an image. An interest pixel is also called a corner, keypoint, or salient image point.

Interest Strength of a Pixel: The magnitude of the change in pixel values along different 2D directions, e.g., horizontal, vertical.

Local Appearance: The intensities of neighboring pixels centered by a given pixel.

Non-Maximum Suppression: A candidate selection scheme that resets the strength of a pixel to zero, i.e., eliminates its candidacy, if it is not a local maximum.

Repeatability: An interest pixel found in one image can be found in another again if these two images are spatially similar to each other.

An Introduction to Kernel Methods

Gustavo Camps-Valls
Universitat de València, Spain

Manel Martínez-Ramón
Universidad Carlos III de Madrid, Spain

José Luis Rojo-Álvarez
Universidad Rey Juan Carlos, Spain

INTRODUCTION

Machine learning has experienced a great advance in the eighties and nineties due to the active research in artificial neural networks and adaptive systems. These tools have demonstrated good results in many real applications, since neither *a priori* knowledge about the distribution of the available data nor the relationships among the independent variables should be necessarily assumed. Overfitting due to reduced training data sets is controlled by means of a regularized functional which minimizes the complexity of the machine. Working with high dimensional input spaces is no longer a problem thanks to the use of kernel methods. Such methods also provide us with new ways to interpret the classification or estimation results. Kernel methods are emerging and innovative techniques that are based on first mapping the data from the original input feature space to a kernel feature space of higher dimensionality, and then solving a linear problem in that space. These methods allow us to geometrically design (and interpret) learning algorithms in the kernel space (which is nonlinearly related to the input space), thus combining statistics and geometry in an effective way. This theoretical elegance is also matched by their practical performance.

Although kernels methods have been considered from a long time ago in pattern recognition from a theoretical point of view (see, e.g., Capon, 1965), a number of powerful kernel-based learning methods emerged in the last decade. Significant examples are Support Vector Machines (SVMs) (Vapnik, 1998), Kernel Fisher Discriminant (KFD), (Mika, Ratsch, Weston, Scholkopf, & Mullers, 1999) Analysis, Kernel Principal Component Analysis (PCA) (Schölkopf, Smola and Müller, 1996), Kernel Independent Component Analysis Kernel (ICA) (Bach and Jordan, 2002), Mutual Information (Gretton, Herbrich, Smola, Bousquet, Schölkopf, 2005), Kernel ARMA (Martínez-Ramón, Rojo-Álvarez, Camps-Valls, Muñoz-Marí, Navia-Vázquez, Soria-Olivas, & Figueiras-Vidal, 2006), Partial Least Squares (PLS) (Momma & Bennet, 2003), Ridge Regression (RR) (Saunders, Gammerman, & Vovk, 1998), Kernel K-means (KK-means) (Camastra, & Verri, 2005), Spectral Clustering (SC) (Szymkowiak-Have, Girolami & Larsen, 2006), Canonical Correlation Analysis (CCA) (Lai & Fyfe, 2000), Novelty Detection (ND) (Schölkopf, Williamson, Smola, & Shawe-Taylor, 1999) and a particular form of regularized AdaBoost (Reg-AB), also known as Arc-GV (Rätsch, 2001). Successful applications of kernel-based algorithms have been reported in various fields such as medicine, bioengineering, communications, data mining, audio and image processing or computational biology and bioinformatics.

In many cases, kernel methods demonstrated results superior to their competitors, and also revealed some additional advantages, both theoretical and practical. For instance, kernel methods (i) efficiently handle large input spaces, (ii) deal with noisy samples in a robust way, and (iii) allow embedding user knowledge about the problem into the method formulation easily. The interest of these methods is twofold. On the one hand, the machine-learning community has found in the kernel concept a powerful framework to develop efficient nonlinear *learning* methods, and thus solving efficiently complex problems (e.g. pattern recognition, function approximation, clustering, source independence, and density estimation). On the other hand, these methods can be easily used and tuned in many research areas, e.g. biology, signal and image processing, communica-

tions, etc, which has also captured the attention of many researchers and practitioners in safety-related areas.

BACKGROUND

Kernel Methods offer a very general framework for machine learning applications (classification, clustering regression, density estimation and visualization) over many types of data (time series, images, strings, objects, etc). The main idea of kernel methods is to embed the data set $S \subseteq X$ into a higher (possibly infinite) dimensional Hilbert space $\mathcal{H}$. The mapping of the data S into the Hilbert Space $\mathcal{H}$ is done through a nonlinear transformation $x \mapsto f(x)$. Thus, there will be a nonlinear relationship between the input data x and its image in $\mathcal{H}$. Then, one can use linear algorithms to detect relations in the embedded data that will be viewed as nonlinear from the point of view of the input data.

This is a key point of the field: using linear algorithms provides many advantages since a well-established theory and efficient methods are available. The mapping is denoted here by $\phi: X \rightarrow \mathcal{H}$, where the Hilbert space $\mathcal{H}$ is commonly known also as *feature space*. Linear algorithms will benefit from this mapping because of the higher dimensionality of the Hilbert space. The computational burden would dramatically increase if one needed to deal with high dimensionality vectors, but there is a useful trick (the *kernel trick*) that allows us to use kernel methods. As a matter of fact, one can express almost any linear algorithm as a function of dot products among vectors. Then, one does not need to work with the vectors once the dot products have been computed. The kernel trick consists of computing the dot products of the data into the Hilbert space $\mathcal{H}$ as a function of the data in the input space. Such a function is called a Mercer's kernel. If it is available, one can implement a linear algorithm into a higher (possibly infinite) Hilbert Space $\mathcal{H}$ without needing to explicitly deal with vectors in these space, but just their dot products. Figure 1 illustrates several kernel methods in the feature spaces. In Figure 1(a), the classical SVM is shown, which basically solves the (linear) optimal separating hyperplane in a high dimensional feature spaces. Figure 1(b) shows the same procedure for the KFD, and Figure 1(c) shows how a novelty detection (known as one-class SVM) can be developed in feature spaces.

The above procedures are done under the framework of the Theorem of Mercer (Aizerman, Braverman & Rozonoér, 1964). A Hilbert space $\mathcal{H}$ is said to be a Reproducing Kernel Hilbert Space (RKHS) with a Reproducing Kernel Inner Product K (often called RKIP or more commonly, Kernel) if the members of $\mathcal{H}$ are functions on a given interval T and if kernel K is defined on the product $T \times T$ having the properties (Aronszajn, 1950):

- for every $t \in T$, $K(\cdot,t) \in \mathcal{H}$, with value at $s \in T$ equal to $K(s,t)$.
- There is a reproducing kernel inner product defined as $(g, K(\cdot,t))_K = g(t)$ for every g in $\mathcal{H}$.

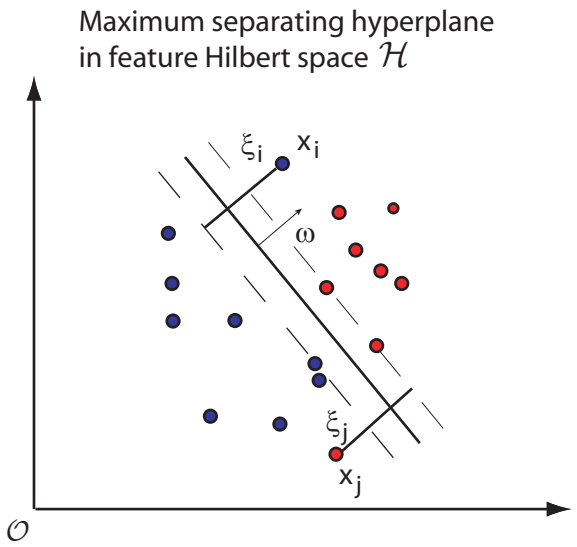
The Mercer's theorem states that there exist a function $\varphi: \mathbb{R}^n \rightarrow \mathcal{H}$ and a dot product $K(s, t) = \langle \varphi(s), \varphi(t) \rangle$ if and only if for any function $g(t)$ for which $\int g(t)dt < \infty$ the inequality $\int K(s, t)g(s)g(t)dsdt \geq 0$ is satisfied.

This condition is not always easy to prove for any function. The first kernels to be proven to fit the Mercer theorem were the polynomial kernel and the Gaussian kernel.

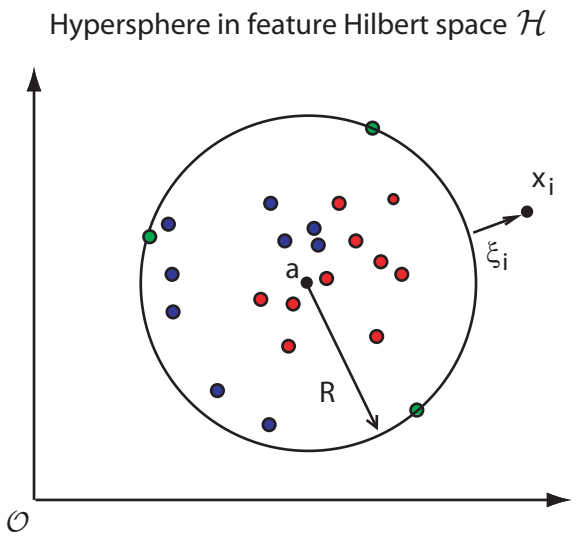
It is worth noting here that mapping ϕ does not require to be explicitly known to solve the problem. In fact, kernel methods work by computing the *similarity* among training samples (the so-called *kernel matrix*) by implicitly measuring distances in the feature space through the pair-wise inner products $\langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle$ between mapped samples $\mathbf{x}, \mathbf{z} \in X$. The matrix $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$ (where $\mathbf{x}_i, \mathbf{x}_j$ are data points) is called the *kernel matrix* and contains all necessary information to perform many (linear) classical algorithms in the embedding space. As we said before, a linear algorithm can be transformed into its non-linear version with the so-called *kernel trick*.

The interested reader can find more information about all these methods in (Vapnik, 1998; Cristianini & Shawe-Taylor, 2000; Schölkopf & Smola, 2002; Shawe-Taylor & Cristianini, 2004). Among all good properties revised before, at present the most active area of research is the design of kernels for specific domains, such as string sequences in bioinformatics, image data, text documents, etc. The website www.kernel-machines.org provides free software, datasets, and constantly updated pointers to relevant literature.

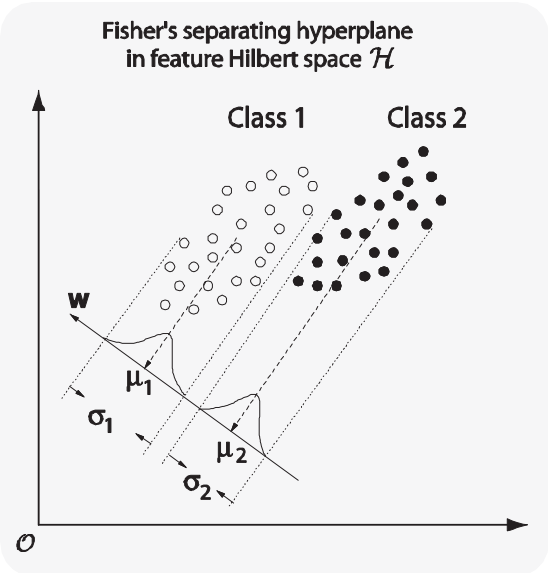
Figure 1.



(a) SVM. Linear decision hyperplane in a non-linearly transformed space, where slack variables ξ_i are included to deal with committed errors.



(b) One-class SVM. The hypersphere containing the (colored) target data is described by the center **a** *and radius* **R,** *in which the samples in the boundary are the support vectors (green) and samples outside the ball are assigned a positive constrained to deal with outliers.*



(c) KFD analysis. Illustration of Fisher's discriminant analysis for two classes. One searches for a direction w such that both the difference between the class means projected onto this directions (μ1, μ2) is large and the variance around these means (μ1, μ2) is small. The kernel Fisher discriminant performs this operation in a kernel space.

FUTURE TRENDS

Despite its few years of conception (early 1990s), the field of kernel methods is a mature enough research area and a huge amount of real-life successful applications have been reported. At present, the field is moving towards more challenging applications, such as working with huge databases, low number of labeled samples, transductive approaches, refined string kernels or the combination and fusion of both different domain knowledge and design of the kernel matrix.

CONCLUSION

This chapter was devoted to the definition of kernel methods and to the revision of the main general methods of application of them. The main feature of kernel methods applied to statistical learning is that they provide nonlinear properties to methods that are formulated in a linear way. This goal is achieved by

the kernel trick, which consists of a mapping of the data from an input space into a RKHS.

RKHS are Hilbert spaces provided with an inner product that is a function of the data in the input space. Therefore, one can compute dot products of the data in the Hilbert space without explicitly knowing the vectors in this space. Using this feature, linear algorithms can be "kernelized", so adding to them nonlinear properties but treating them as linear in these spaces implicitly. A set of references on the most known kernelized methods has been provided in this chapter.

REFERENCES

Aizerman, M. A., Braverman, É . M., & Rozonoér, L. I. (1964). Theoretical foundations of the potential function method in pattern recognition learning, *Automation and Remote Control*, 25, 821–837.

Aronszajn, N. (1950). Theory of reproducing kernels. *Trans. Amer. Math. Soc.*, 68, 337-404.

Bach, F. R. and Jordan, M. I. (2002). Kernel independent component analysis. *Journal of Machine Learning Research, 3*, 1-48.

Capon, J. (1965). Hilbert space method for detection theory and pattern recognition. *IEEE Transactions on Information Theory, 11*(2), 247-259.

Camastra, F., & Verri, A. (2005). A Novel Kernel Method for Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5).

Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to support vector machines*. Cambridge, U.K.: Cambridge University Press.

Gretton, A., Herbrich, R., Smola, A., Bousquet, & O., Schölkopf, B. (2005). Kernel Methods for Measuring Independence, *Journal of Machine Learning Research*, Vol. 6, pp. 2075-2129.

Lai, P. L. & Fyfe, C. (2000). Kernel and nonlinear canonical correlation analysis, Int. J. Neural Syst., vol. 10, no. 5, pp. 365–377.

Martínez-Ramón, M., Rojo-Álvarez, J. L., Camps-Valls, G., Muñoz-Marí, J., Navia-Vázquez, A., Soria-Olivas, E., & Figueiras-Vidal, A. (2006) Support Vector Machines for Nonlinear Kernel ARMA System Identification, *IEEE Transactions on Neural Networks*.

Mika, S., Ratsch, G., Weston, J., Scholkopf, B., & Mullers, K.R. (1999). *Fisher discriminant analysis with kernels*, IEEE Signal Proc. Society Workshop in Neural Networks for Signal Processing IX, pp. 41-48.

Momma, M & Bennet, K. (2003), Kernel Partial Least Squares, in *Proc. of the Conference on Learning Theory (COLT 2003)*, pp. 216-230.

Rätsch, G. "Robust boosting via convex optimization," Ph.D. dissertation, Univ. Potsdam, Potsdam, Germany, Oct. 2001. [Online]. Available: http://www.boosting.org/papers/thesis.ps.gz.

Saunders, C., Gammerman, A., & Vovk, V. (1998). Ridge regression learning algorithm in dual variables, *Proceedings of the 15th International Conference on Machine Learning ICML-98*, Madison-Wisconsin, pp. 515-521.

Schölkopf, B. (1997). *Support Vector Learning*. R. Oldenbourg Verlag, Munich.

Schölkopf, B., & Smola, A. (2002). *Learning with kernels*. Cambridge, MA: MIT Press.

Schölkopf, B., Smola, A. & Müller, K.-R. (1996), *Nonlinear Principal Component Analysis as a Kernel Eigenvalue Problem*, Technical Report 44, Max Planck Institut für Biologische Kybernetik, Tübingen, Germany, Dec. 1996.

Schölkopf, B., Williamson R. C., Smola, A. & Shawe-Taylor, J. (1999), Support Vector Method for Novelty Detection, *Advances in Neural Information Processing Systems 12*, Denver, CO.

Shawe-Taylor, J., & Cristianini, N. (2004). *Kernel methods for pattern analysis.* Cambridge University Press.

Szymkowiak-Have, A., Girolami, M.A. & Larsen, J. (2006). Clustering via kernel decomposition, IEEE Transactions on Neural Networks, Vol. 17, No 1, pp. 256 – 264.

Vapnik, V. (1998). *Statistical learning theory.* New York: Wiley.

KEYTERMS

Hilbert Space: A Hilbert space is a generalization of the Euclidean space not restricted to finite dimensions, thus being an inner product space.

Kernel Trick: This is a method for converting a linear algorithm into a non-linear one by mapping the samples in the input space to a higher-dimensional (possibly infinite) space so that solving a linear problem is more likely. This linear solution is non-linear in the original input space. This can be done using Mercer's condition, which states that any positive semi-definite kernel $K(x,y)$ can be expressed as a dot product in a high-dimensional space.

Maximum-Margin Hyperplane: In the context of geometry, this is a hyperplane that separates two sets of points, being at the same time at equal and maximum distance from the two.

Positive-Definite Matrix. An $n \times n$ Hermitian matrix M fulfilling that all its eigenvalues λ_i are positive or equivalently, for all non-zero (complex) vectors $\mathbf{z}$, $\mathbf{z}^*\mathbf{Mz} > 0$ where $\mathbf{z}^*$ indicates the conjugate transpose of $\mathbf{z}$.

Quadratic Programming (QP). This is a special type of mathematical optimization problem in which, given a sample $\mathbf{x}$ in $\mathbb{R}^n$, the $n \times n$ matrix Q is symmetric, and an $n \times 1$ vector c, the problem consists of minimizing with respect to $\mathbf{x}$:

$$f(x) = \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{c}^T\mathbf{x}$$

where superscript T is the transpose operator.

Regularization: In order to provide smoothness to the solution of an inverse problem, the problem must be regularized usually by imposing a bound on the norm of the model weights, inducing a natural trade-off between fitting the data and reducing a norm of the solution.

Reproducing Kernel Hilbert Space is a function space in which point-wise evaluation is a continuous linear functional, i.e. they are spaces that can be defined by reproducing kernels. Formally, given a dataset X and H a Hilbert space of (potentially) complex-valued functions on X, then H is a reproducing kernel Hilbert space if and only if the linear map $x \mapsto f(x)$ from H to the complex numbers is continuous for any x in X. By the Riesz representation theorem, this implies that for given x there exists an element K_x of H with the property that: $f(x) = \langle K_x, f \rangle \; \forall f \in H$. The function $K(x, y) \equiv K_x(y)$ is called a reproducing kernel for the Hilbert space.

Structural Risk Minimization (SRM). In contrast with empirical risk minimization, the SRM is to find the learning machine that yields a good trade-off between low empirical risk and small capacity (*see* regularization).

The Issue of Missing Values in Data Mining

Malcolm J. Beynon
Cardiff University, UK

INTRODUCTION

The essence of data mining is to investigate for pertinent information that may exist in data (often large data sets). The immeasurably large amount of data present in the world, due to the increasing capacity of storage media, manifests the issue of the presence of missing values (Olinsky *et al*., 2003; Brown and Kros, 2003). The presented encyclopaedia article considers the general issue of the presence of missing values when data mining, and demonstrates the effect of when managing their presence is or is not undertaken, through the utilisation of a data mining technique.

The issue of missing values was first exposited over forty years ago in Afifi and Elashoff (1966). Since then it is continually the focus of study and explanation (El-Masri and Fox-Wasylyshyn, 2005), covering issues such as the nature of their presence and management (Allison, 2000). With this in mind, the naïve consistent aspect of the missing value debate is the limited general strategies available for their management, the main two being either the simple deletion of cases with missing data or a form of imputation of the missing values in someway (see Elliott and Hawthorne, 2005). Examples of the specific investigation of missing data (and data quality), include in; data warehousing (Ma *et al*., 2000), and customer relationship management (Berry and Linoff, 2000).

An alternative strategy considered is the retention of the missing values, and their subsequent 'ignorance' contribution in any data mining undertaken on the associated original incomplete data set. A consequence of this retention is that full interpretability can be placed on the results found from the original incomplete data set. This strategy can be followed when using the nascent CaRBS technique for object classification (Beynon, 2005a, 2005b). CaRBS analyses are presented here to illustrate that data mining can manage the presence of missing values in a much more effective manner than the more inhibitory traditional strategies. An example data set is considered, with a noticeable level of missing values present in the original data set. A critical increase in the number of missing values present in the data set further illustrates the benefit from 'intelligent' data mining (in this case using CaRBS).

BACKGROUND

Underlying the necessity to concern oneself with the issue of missing values is the reality that most data analysis techniques were not designed for their presence (Schafer and Graham, 2002). It follows, an external level of management of the missing values is necessary. There is however underlying caution on the ad-hoc manner in which the management of missing values may be undertaken, this lack of thought is well expressed by Huang and Zhu (2002, p. 1613):

Inappropriate treatment of missing data may cause large errors or false results.

A recent article by Brown and Kros (2003) looked at the impact of missing values on data mining algorithms, including; *k*-nearest neighbour, decision trees, association rules and neural networks. For these considered techniques, the presence of missing values is considered to have an impact, with a level external management necessary to accommodate them. Indeed, perhaps the attitude is that it is the norm to have to manage the missing values, with little thought to the consequences of doing this. Conversely, there is also the possibilities that missing values differ in important ways from those that are present.

While common, the specific notion of the management of missing values is not so clear, since firstly it is often necessary to understand the reasons for their presence (De Leeuw, 2001), and subsequently how these reasons may dictate how they should be future described. For example, in the case of large survey data, whether the missing data is (*ibid*.); Missing by design, Inapplicable item, Cognitive task too difficult, Refuse to respond, Don't know and Inadequate score. Whether the data is survey based or from another source, a typi-

cal solution is to make simplifying assumptions about the mechanism that causes the missing data (Ramoni and Sebastiani, 2001). These mechanisms (causes) are consistently classified into three categories, based around the distributions of their presence, namely;

- *Missing Completely at Random (MCAR)*: The fact that an entry is missing is independent of both observed and unobserved values in the data set (e.g. equipment failure).
- *Missing at Random (MAR)*: The fact that an entry is missing is a function of the observed values in the data set (e.g. respondents are excused filling in part of a questionnaire).
- *Missing Not at Random (MNAR)*: An entry will be missing depends on both observed and unobserved values in the data set (e.g. personal demographics of a respondent contribute to the incompleteness of a questionnaire).

Confusion has surrounded the use of these terms (Regoeczi and Riedel, 2003). Further, the type of missing data influences the available methods to manage their presence (Elliott and Hawthorne, 2005). Two most popular approaches to their management are case deletion and imputation (next discussed), in the cases of the missing values being MCAR or MAR then imputation can produce a more complete data set that is not adversely biased (*ibid*.).

The case deletion approach infers cases in a data set are discarded if their required information is incomplete. This, by its nature incurs the loss of information from discarding partially informative case (Shen and Lai, 2001), De Leeuw (2001) describes the resultant loss of information, less efficient estimates and statistical tests. Serious biases may be introduced when missing values are not randomly distributed (Huang and Zhu, 2002). A further problem occurs if there is a small sample so deletion of too many cases may reduce the statistical significance of conclusions. Also associated with this approach is re-weighting, whereby the remaining complete cases can be weighted so that their distribution more closely resembles that of the full sample (Schafer and Graham, 2002; Huang and Zhu, 2002).

Imputation infers an incomplete data set becomes filled-in by the replacement of missing values with surrogates (Olinsky et al., 2003). It is potentially more efficient than case deletion, because no cases are sacrificed, retaining the full sample helps to prevent loss of power resulting from a diminished sample size (Schafer and Graham, 2002). De Leeuw (2001) identifies the availability of modern and user-friendly software encourages the use of imputation, with approaches that include; hot deck imputation, cold deck imputation, multiple imputation, regression and stochastic regression imputation.

Beyond these approaches the commonest is mean imputation, whereby the missing value for a given attribute in a case is filled in with the mean of all the reported values for that attribute (Elliott and Hawthorne, 2005). One factor often highlighted with the utilisation of mean imputation is that the distribution characteristics (including variance) of the completed data set may be underestimated (El-Masri and Fox-Wasylyshyn, 2005). The accompanying danger here is that this approach lulls the user into the plausible state of believing that the data are complete after all. The possible effectiveness of mean imputation, and other approaches, depends of course on the level of incompleteness inherent, with many studies testing results with differing percentages of missing data present (ibid.).

MAIN THRUST

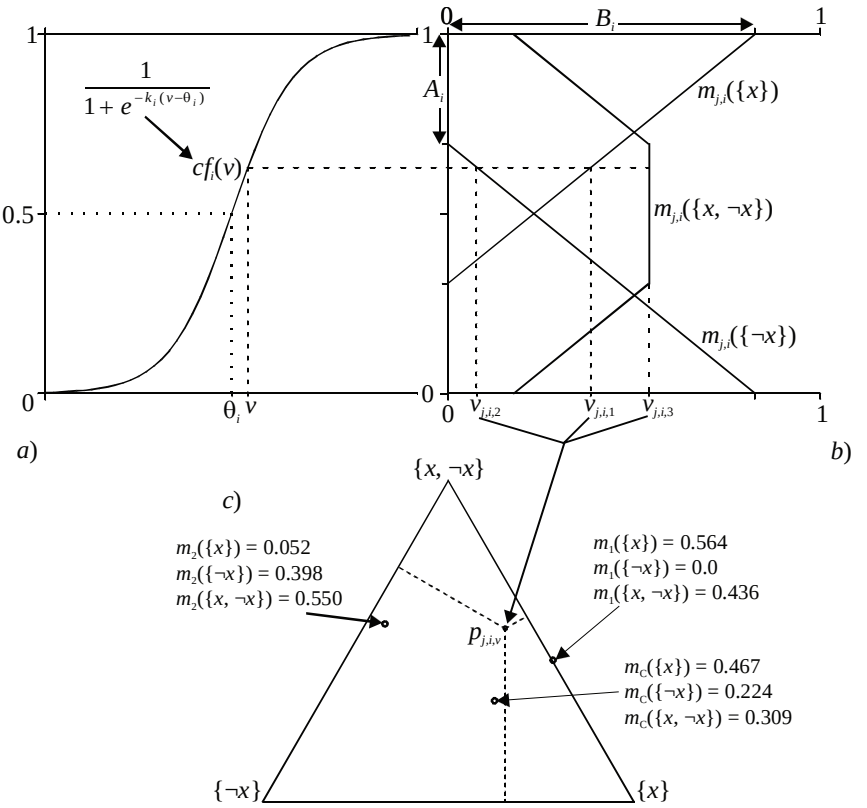
To demonstrate the effect of the management of missing values in data mining, the nascent CaRBS technique is employed (for a detailed elucidation see Beynon, 2005a, 2005b). The aim of the CaRBS technique is to construct a body of evidence (BOE) for each characteristic value that purports levels of exact belief (mass values – $m_{j,i}(\cdot)$) towards the classification of an object to a given hypothesis ($m_{j,i}(\{x\})$), its complement ($m_{j,i}(\{\neg x\})$) and concomitant ignorance ($m_{j,i}(\{x, \neg x\})$). More formally, the mass values come from:

$$m_{j,i}(\{x\}) = \frac{B_i}{1-A_i} cf_i(v) - \frac{A_i B_i}{1-A_i},$$

$$m_{j,i}(\{\neg x\}) = \frac{-B_i}{1-A_i} cf_i(v) + B_i,$$

and $m_{j,i}(\{x, \neg x\}) = 1 - m_{j,i}(\{x\}) - m_{j,i}(\{\neg x\})$, where $cf_i(v)$ is a confidence value (usually sigmoid function). A graphical representation of this process is given in Figure 1.

Figure 1. Stages within the CaRBS technique for a characteristic value $v_{j,i}$



The stages shown in Figure 1 report the construction and final representation in a simplex plot of a characteristic BOE $m_{j,i}(\cdot)$. A series of characteristic BOEs are then combined to form an object BOE that offers the final classification of the respective object (to x or $\neg x$), using Dempster's combination rule. A graphical example of this combination process is shown in Figure 1*c* ($m_1(\cdot)$ and $m_2(\cdot)$ combining to make $m_C(\cdot)$).

Within CaRBS, if a characteristic value is missing its characteristic BOE supports only ignorance, namely $m_{j,i}(\{x, \neg x\}) = 1$ (with $m_{j,i}(\{x\}) = m_{j,i}(\{\neg x\}) = 0$). The consequence is, a missing value is considered an ignorant value and so offers no evidence in the subsequent classification of an object, allowing their retention in the analysis (see Beynon, 2005b).

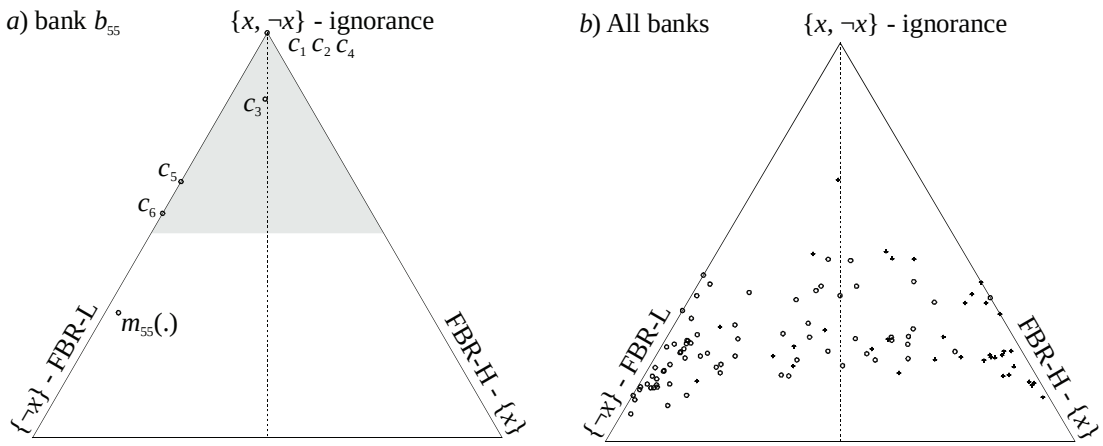
The Fitch Bank Individual Ratings (FBRs) on 108 UK banks is considered (FitchRatings, 2003), partitioned into two groups discerning between 'more than a strong bank' (FBR-H) and 'no more than a strong bank' (FBR-L). The FBR data set was obtained from the Bankscope database (Bankscope, 2005), with 14.35% of the data originally missing. Following Pasiouras *et al*. (2005), six financial characteristics describe the UK banks, briefly abbreviated here to; c_1 (42), c_2 (0), c_3 (7), c_4 (37), c_5 (7) and c_6 (0). The values in brackets indicate the number of missing values of each characteristic within the 108 banks.

The first analysis is undertaken on the original incomplete FBR data set. A configured CaRBS system optimises the classification of the banks through their object BOEs, to either FBR-H or FBR-L, by only minimising ambiguity (see Beynon, 2005b). The defined constrained optimisation problem was solved using trigonometric differential evolution (see Fan and Lampinen, 2003). The results for one bank are next presented, namely b_{55}, known to be classified to FBR-L (its description includes two missing values, c_1 and c_4). Presented in Table 1, are the characteristic BOEs $m_{55,i}(\cdot)$, $i = 1, \ldots, 6$, that describe the evidence from all the characteristics to the bank's FBR classification (for their evaluation see Beynon, 2005a).

Table 1. Characteristic BOEs $m_{55,i}(\cdot)$, $i = 1, \ldots, 6$ for the bank b_{55}

BOEs	c_1	c_2	c_3	c_4	c_5	c_6
Financial values	-	6.65	0.82	-	40.43	6.66
$m_{55,?}$({FBR-H})	0.000	0.000	0.078	0.000	0.000	0.000
$m_{55,?}$({FBR-L})	0.000	0.000	0.086	0.000	0.368	0.446
$m_{55,?}$({FBR-H, FBR-L})	1.000	1.000	0.836	1.000	0.632	0.554

Figure 2. Simplex plots of the FBR classification of the bank b_{55} and all banks



In Table 1 the missing characteristics, c_1 and c_4, offer only ignorant evidence ($m_{55,i}$({FBR-H, FBR-L}) = 1.000), as does c_2 (through the optimisation). For the bank b_{55}, these characteristic BOEs can be combined to construct its object BOE (using Dempster's combination rule), giving; m_{55}({FBR-H}) = 0.029, m_{55}({FBR-L}) = 0.663 and m_{55}({FBR-H, FBR-L}) = 0.309. The evidence towards each bank's FBR classification can be presented using the simplex plot method of data representation, see Figure 2.

In the simplex plot in Figure 2*a*, the base vertices identify certainty in the classification of a bank to either FBR-L (left) or FBR-H (right), and the top vertex is associated with total ignorance. The circles positioned in the grey-shaded region represent characteristic BOEs ($m_{55,i}(\cdot)$), with the resultant object BOE labelled $m_{55}(\cdot)$. The vertical dashed line identifies between where the evidence and final classification are more towards one FBR classification over the other, the bank b_{55} is shown correctly classified to FBR-L. In Figure 2*b*, the final FBR classifications of the 108 banks are reported using their object BOEs (FBR-L and FBR-H banks signified by circles and crosses), indicating 77.778% classification accuracy.

The next CaRBS analysis on the incomplete FBR data set includes when the missing values are managed using mean imputation. Utilising the simplex plot representation of the results, the FBR classification of bank b_{55} and final classification of all banks are given, see Figure 3.

For the bank b_{55} described in Figure 3*a*, the interesting feature is the contribution of c_1, which here offers incorrect evidence towards the bank's classification, even though it was a missing value that has been assigned a surrogate value. From Figure 3*b*, there is 75.926% classification accuracy, lower than that found earlier.

Figure 3. FBR classification of the bank b_{55} and all banks (missing values imputed)

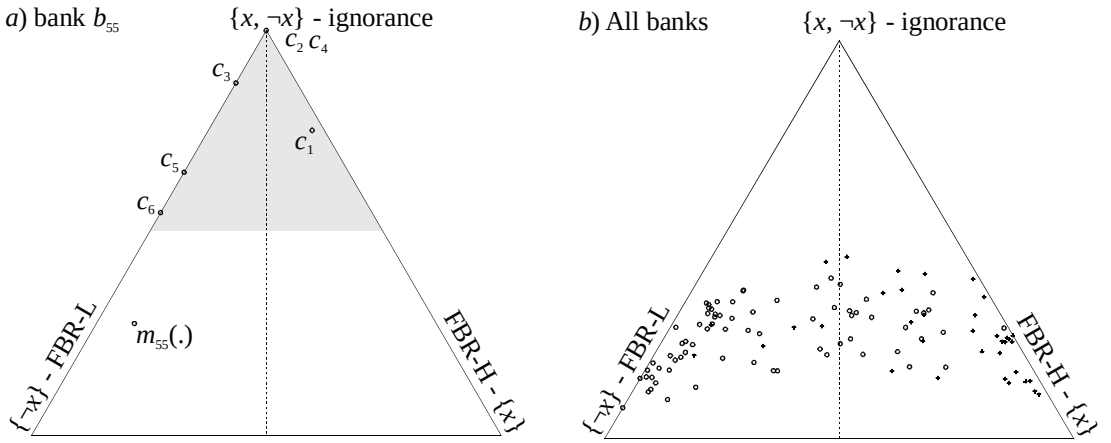


Table 2. Characteristic BOEs $m_{55,i}(\cdot)$, i = 1, …, 6 for the bank b_{55}

BOEs	c_1	c_2	c_3	c_4	c_5	c_6
Financial values	-	6.65	-	-	-	-
$m_{55,?}$({FBR-H})	0.000	0.000	0.000	0.000	0.000	0.000
$m_{55,?}$({FBR-L})	0.000	0.180	0.000	0.000	0.000	0.000
$m_{55,?}$({FBR-H, FBR-L})	1.000	0.820	1.000	1.000	1.000	1.000

The next CaRBS analysis considers a more incomplete FBR data set, with only 25% of the original data retained (no bank has all their characteristics defined missing). This level of incompleteness is critical to effective data mining (see Background section), but a CaRBS analysis can still be performed on it without any concern. For the bank b_{55}, it now only has one characteristic value not missing (c_2), the respective characteristic BOEs associated with it are reported in Table 2.

From Table 2, for the bank b_{55} the resultant object BOE is; $m_{55}(\{x\}) = 0.000$, $m_{55}(\{\neg x\}) = 0.180$ and $m_{55}(\{x, \neg x\}) = 0.820$ (the same as $m_{55,2}(\cdot)$), see also Figure 4.

In Figure 4*a*, the position of the object BOE $m_{55}(\cdot)$ for b_{55} infers more ignorance associated with its classification than in the previous analyses, a direct consequence of the majority of its characteristic values now being missing. In Figure 4*b*, a number of the presented object BOEs are nearer the top vertex than in the previous cases, again due to the large number of missing values present (a 83.333% classification accuracy exists in this case). Importantly, since no transformation of the data set has been undertaken, these results are interpretable. In the next analysis, all missing values are replaced using mean imputation (see Figure 5).

For the bank b_{55}, in Figure 5*a*, the c_1, c_3 and c_6 values contribute evidence even though they are all surrogates for missing values, whereas c_2 that was not missing now contributes only ignorance (Figure 5*b* identifies a 81.481% classification accuracy). This demonstrates how dramatically different results can be found when undertaking a large amount of imputation in a data set.

Figure 4. FBR classification of the bank b_{55} and all banks (25% available data)

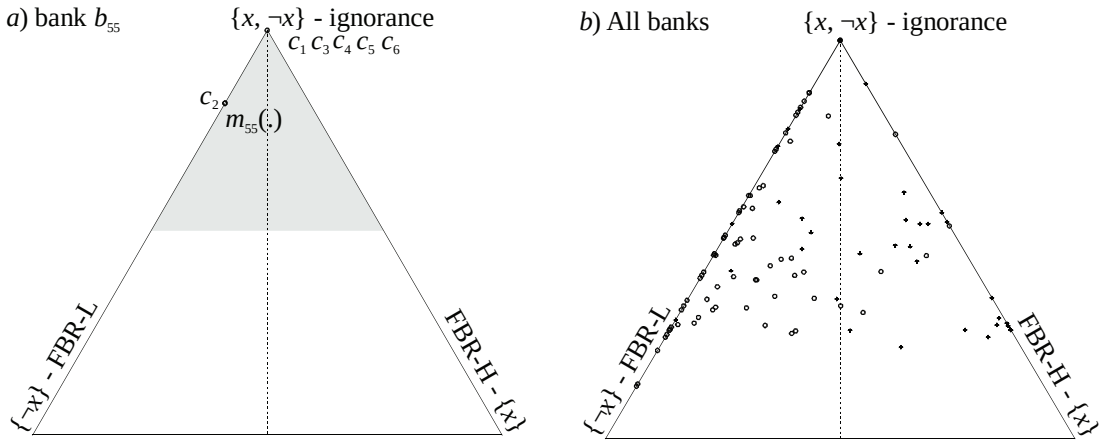
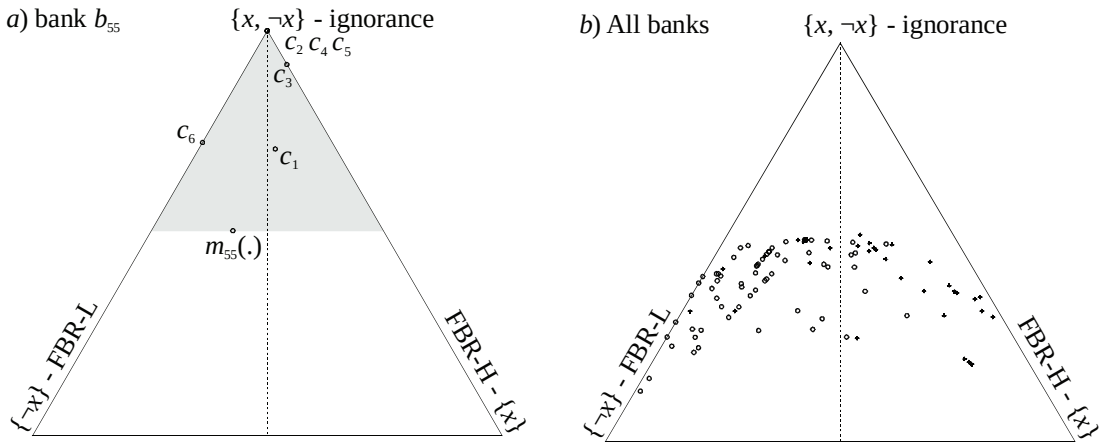


Figure 5. FBR classification of the bank b_{55} and all banks (25% available data the rest imputed)



FUTURE TRENDS

Now, and in the future, the issue of missing values in data mining is not going to go away. Indeed, the concomitant literature continually suggests its increased presence, a facet of the increased capacity of computing power and storage media. Consequently, data mining techniques have to continue to offer the most realistic analyses in their presence.

For the most part, their presence is externally managed using approaches that still follow one of two main strategies, namely case deletion or imputation. Modern techniques like CaRBS show the way forward to their internal management. The future may require that the more traditional mining techniques should attempt to follow how the modern techniques work in the presence of missing values.

CONCLUSION

A general view of data mining is that it involves analysing data that the analyst probably had no direct control on its gathering. Moreover, it may be secondary data

accrued from one or more sources. The implication here is that the presence of missing values is similarly out of the control of the analyst, hence the need for their management. However, almost without question, the management has meant the alteration of the considered data set.

The CaRBS technique illustrates one example where it is possible to analyse data with missing values present without any need to incumbently manage them. The level of difference in the analyses when case deletion or imputation are or are not undertaken demonstrates the potential costs that need to be realised when managing missing values. This latter fact is perhaps most concerning since it is probably true that interpretability to data mining results are often undertaken even when missing values have been crudely managed (with the management action performed then soon forgotten).

REFERENCES

Afifi, A.A., & Elashoff, R. (1966). Missing observations in multivariate statistics. Part 1: Review of the literature, *Journal of the American Statistical Association*, 61, 595-604.

Allison, P.D. (2000). *Missing Data*, Sage University Paper Series, Quantitative Applications in the Social Sciences, 136. Thousand Oaks: Sage.

Bankscope (2005). http://www.bvdep.com/bankscope.html, Accessed 06/12/05.

Berry, M., & Linoff, G. (2000). The art and science of customer relationship, *Industrial Management & Data Systems*, 100(5), 245-246.

Beynon, M.J. (2005a). A Novel Technique of Object Ranking and Classification under Ignorance: An Application to the Corporate Failure Risk Problem, *European Journal of Operational Research*, 167(2), 493-517.

Beynon, M.J. (2005b). A Novel Approach to the Credit Rating Problem: Object Classification Under Ignorance, *International Journal of Intelligent Systems in Accounting, Finance and Management*, 13, 113-130.

Brown, M.L., & Kros, J.F. (2003). Data mining and the impact of missing data, *Industrial Management & Data Systems*, 103(8), 611-621.

DeLeeuw, E.D. (2001). Reducing Missing Data in Surveys: An Overview of Methods, *Quality & Quantity*, 35, 147-160.

El-Masri, M.M., & Fox-Wasylyshyn, S.M. (2005). Missing Data: An Introductory Conceptual Overview for the Novice Researcher, *Canadian Journal of Nursing Research*, 37(4), 156-171.

Elliott, P., & Hawthorne, G. (2005). Imputing missing repeated measures data: how should we proceed?, *Australian and New Zealand Journal of Psychiatry*, 39, 575-582.

Fan, H.-Y., & Lampinen, J. (2003). A Trigonometric Mutation Operation to Differential Evolution. *Journal of Global Optimization*, 27, 105-129.

FitchRatings (2003). Launch of Fitch's Bank Support Rating methodology, www.fitchrating.com, accessed on 17/12/05.

Huang, X., & Zhu, Q. (2002). A pseudo-nearest-neighbour approach for missing data on Gaussian random data sets, *Pattern Recognition Letters*, 23, 1613-1622.

Ma, C., Chou, D., & Yen, D. (2000). Data warehousing, technology assessment and management, *Industrial Management & Data Systems*, 100(3), 125-135.

Olinsky, A., Chen, S., & Harlow, L. (2003). The comparative efficacy of imputation methods for missing data in structural equation modelling, *European Journal of Operational Research*, 151, 53-79.

Pasiouras, F., Gaganis, C., & Zopoundis, C. (2005). A multivariate analysis of Fitch's individual bank ratings, *4th Conference of the Hellenic Finance and Accounting Association*, Piraes, Greece, December.

Ramoni, M. & Sebastiani, P. (2001). Robust Learning with Missing Data, *Machine Learning*, 45, 147-170.

Regoeczi, W.C., & Riedel, M. (2003). The Application of Missing Data Estimation Models to the problem of Unknown Victim/Offender Relationships in Homicide Cases, *Journal of Quantitative Criminology*, 19(2), 155-183.

Schafer, J.L., & Graham, J.W. (2002). Missing Data: Our View of the State of the Art, *Psychological Methods*, 7(2), 147-177.

Shen, S.M., & Lai, Y.L. (2001). Handling Incomplete Quality-of-Life Data, *Social Indicators Research*, 55, 121-166.

KEY TERMS

Body of Evidence (BOE): Collection of mass values and concomitant sets of hypotheses.

CaRBS: Classification and Ranking Belief Simplex.

Case Deletion: A technique for the management of missing values through the removal of a case (object) when any of its characteristic values are missing.

Ignorance: A general term encompassing uncertainty and incompleteness in data analysis.

Imputation: A technique for the management of missing values through the replacement of missing values by calculated surrogates.

Mass Value: The level of exact belief assigned to a set of hypotheses.

Missing Value: A value not present that should describe a characteristic for a case (object).

Simplex Plot: Equilateral triangle domain representation of triplets of non-negative values which sum to one.

Uncertain Reasoning: The attempt to represent uncertainty and reason about it when using uncertain knowledge and imprecise information.

Knowledge Acquisition from Semantically Heterogeneous Data

Doina Caragea
Kansas State University, USA

Vasant Honavar
Iowa State University, USA

INTRODUCTION

Recent advances in sensors, digital storage, computing and communications technologies have led to a proliferation of autonomously operated, geographically distributed data repositories in virtually every area of human endeavor, including e-business and e-commerce, e-science, e-government, security informatics, etc. Effective use of such data in practice (e.g., building useful predictive models of consumer behavior, discovery of factors that contribute to large climatic changes, analysis of demographic factors that contribute to global poverty, analysis of social networks, or even finding out what makes a book a bestseller) requires accessing and analyzing data from multiple heterogeneous sources.

The Semantic Web enterprise (Berners-Lee et al., 2001) is aimed at making the contents of the Web machine interpretable, so that heterogeneous data sources can be used together. Thus, data and resources on the Web are annotated and linked by associating meta data that make explicit the ontological commitments of the data source providers or, in some cases, the shared ontological commitments of a small community of users.

Given the autonomous nature of the data sources on the Web and the diverse purposes for which the data are gathered, in the absence of a universal ontology it is inevitable that there is no unique global interpretation of the data, that serves the needs of all users under all scenarios. Many groups have attempted to develop, with varying degrees of success, tools for flexible integration and querying of data from semantically disparate sources (Levy, 2000; Noy, 2004; Doan, & Halevy, 2005), as well as techniques for discovering semantic correspondences between ontologies to assist in this process (Kalfoglou, & Schorlemmer, 2005; Noy and Stuckenschmidt, 2005). These and related advances in Semantic Web technologies present unprecedented opportunities for exploiting multiple related data sources, each annotated with its own meta data, in discovering useful knowledge in many application domains.

While there has been significant work on applying machine learning to ontology construction, information extraction from text, and discovery of mappings between ontologies (Kushmerick, et al., 2005), there has been relatively little work on machine learning approaches to knowledge acquisition from data sources annotated with meta data that expose the structure (schema) and semantics (in reference to a particular ontology).

However, there is a large body of literature on distributed learning (see (Kargupta, & Chan, 1999) for a survey). Furthermore, recent work (Zhang et al., 2005; Hotho et al., 2003) has shown that in addition to data, the use of meta data in the form of ontologies (class hierarchies, attribute value hierarchies) can improve the quality (accuracy, interpretability) of the learned predictive models.

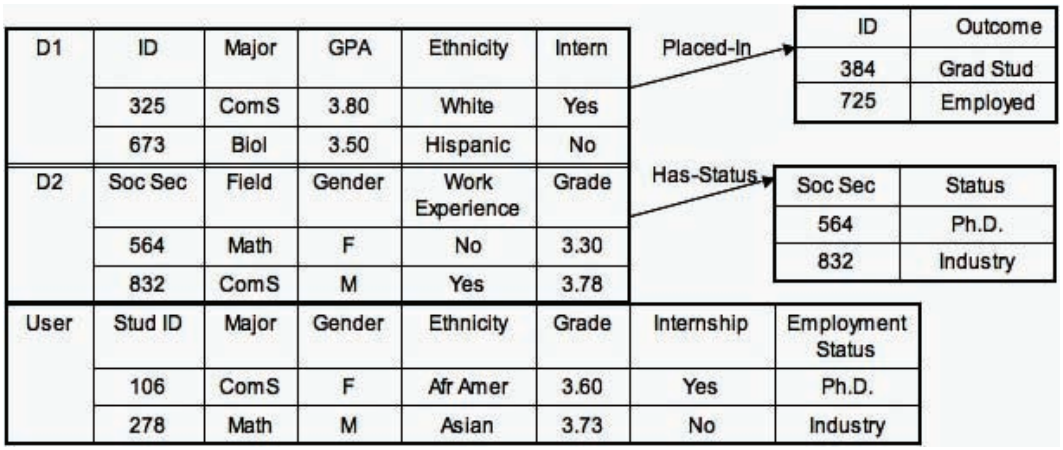
The purpose of this chapter is to precisely define the problem of knowledge acquisition from semantically heterogeneous data and summarize recent advances that have led to a solution to this problem (Caragea et al., 2005).

BACKGROUND

Motivating Example

The problem addressed is best illustrated by an example. Consider two academic departments that independently collect information about their students (Figure 1). Suppose a data set D_1 collected by the first department is organized in two tables, *Student* and *Outcome*, linked by a *Placed-In* relation using

Figure 1. Student data collected by two departments from a statistician's perspective



ID as the common key. Students are described by *ID*, *Major*, *GPA*, *Ethnicity* and *Intern.* Suppose a data set D_2 collected by the second department has a *Student* table and a *Status* table, linked by *Has-Status* relation using *Soc Sec* as the common key. Suppose *Student* in D_2 is described by the attributes *Student ID*, *Field*, *Gender*, *Work-Experience* and *Grade.*

Consider a user, e.g., a university statistician, interested in constructing a predictive model based on data from two departments of interest from his or her own perspective, where the representative attributes are *Student ID*, *Major*, *Gender*, *Ethnicity*, *Grade*, *Internship* and *Employment Status.* For example, the statistician may want to construct a model that can be used to infer whether a typical student (represented as in the entry corresponding to D_U in Figure 1) is likely go on to get a *Ph.D.* This requires the ability to perform queries over the two data sources associated with the departments of interest from the user's perspective (e.g., *fraction of students with internship experience that go onto Ph.D*). However, because the structure (schema) and data semantics of the data sources differ from the statistician's perspective, he must establish the correspondences between the user attributes and the data source attributes.

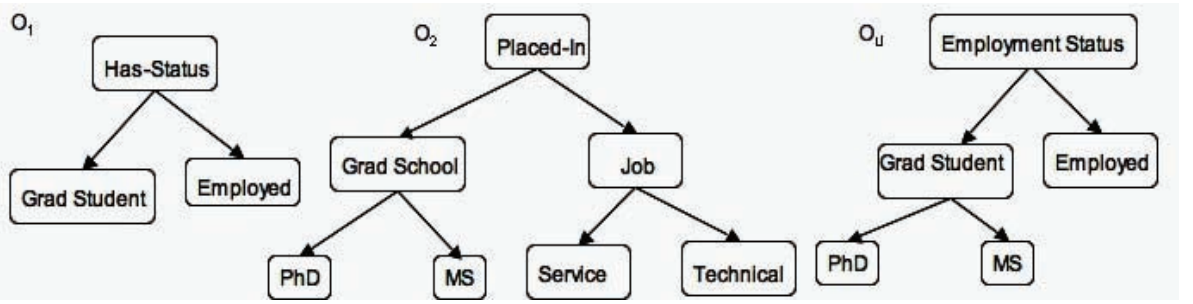
Ontology-Extended Data Sources and User Views

In our framework each data source has associated with it a data source description (i.e., the schema and ontology of the data source). We call the resulting data sources, *ontology extended data sources* (OEDS). An OEDS is a tuple $\mathscr{D}=\{D,S,O\}$, where D is the actual data set in the data source, S the data source schema and O the data source ontology (Caragea et al., 2005). The formal semantics of OEDS are based on ontology-extended relational algebra (Bonatti et al., 2003).

A *data set D* is an instantiation $\mathscr{I}(S)$ of a schema. The *ontology O* of an OEDS $\mathscr{D}$ consists of two parts: *structure ontology*, O_S, that defines the semantics of the data source schema (entities and attributes of entities that appear in data source schema *S*); and *content ontology*, O_I, that defines the semantics of the data instances (values and relationships between values that the attributes can take in instantiations of schema *S*). Of particular interest are ontologies that take the form of *is-a* hierarchies and *has-part* hierarchies. For example, the values of the *Status* attribute in data source D_2 are organized into an *is-a* hierarchy.

Because it is unrealistic to assume the existence of a single global ontology that corresponds to a universally agreed upon set of ontological commitments for all users, our framework allows each user or a community of users to select the ontological commitments that they deem useful in a specific context. A *user's view of data sources* $\mathscr{D}_1,\mathscr{D}_2 \ldots \mathscr{D}_n$ *is specified* by user schema S_U, user ontology O_U, together with a set of semantic *correspondence constraints IC*, and the associated set of *mappings* from the user schema S_U to the data source schemas $S_1,\ldots,S_n$ and from user ontology O_U to the data source ontologies $O_1,\ldots,O_n$ (Caragea et al, 2005).

Figure 2. Attribute value taxonomies (ontologies) O1 and O2 associated with the attributes Has-Status and Placed-In in two data sources of interest. OU is the ontology for Employment Status from the user's perspective



We consider the following types of semantic correspondence constraints: $x \leq y$ (x is semantically subsumed by y), $x \geq y$ (x semantically subsumes y), $x = y$ (x is semantically equivalent to y), $x \neq y$ (x is semantically incompatible with y), $x \approx y$ (x is semantically compatible with y).

Figure 2 shows examples of ontologies that take the form of is-a hierarchies over attribute values. Figure 3 shows some simple examples of user-specified semantic correspondence constraints between the user perspective and the data sources $\mathscr{D}_1$ and $\mathscr{D}_2$ (respectively).

Let $O_1,\ldots,O_n$ be a set of ontologies associated with the data sources $D_1,\ldots,D_n$, respectively, and $P_U=(O_U,IC)$ a user perspective with respect to these ontologies. We say that the ontologies $O_1,\ldots,O_n$, are *integrable* according to the user ontology O_U in the presence of semantic correspondence constraints *IC* if there exist *n* partial injective mappings $\Psi(O_U, O_1),\ldots,\Psi(O_U, O_n)$ from $O_1,\ldots,O_n$, respectively, to O_U.

Related Work

Hull (1997), Davidson et al. (2001), Eckman (2003), Doan & Halevy (2005) survey alternative approaches to data integration, including multi-database systems, mediator based approaches, etc. These efforts addressed, and to varying degrees, solved the following problems in data integration: design of query languages and rules for decomposing queries into sub queries and composing the answers to sub queries into answers to the initial query through schema integration.

However, neither of the existing data integration systems currently support learning from semantically heterogeneous distributed data without first assembling a single data set. While it is possible to retrieve the *data* necessary for learning from a set of heterogeneous data sources, store the retrieved data in a local database, and then apply standard (centralized) learning algorithms, such approach is not feasible when the amounts of data involved are large, and bandwidth and memory are limited, or when the query capabilities of the data sources are limited to providing statistical summaries (e.g., counts of instances that satisfy certain constraints on the values of their attributes) as opposed to data instances. Therefore, solutions to the problem of learning from semantically heterogeneous data sources are greatly needed.

Figure 3. An example of user-specified semantic correspondences between the user ontology OU and data source ontologies O1 and O2 (from Figure 2)

$O_1 \rightarrow O_U$	$O_2 \rightarrow O_U$
ID:O_1=Stud ID:O_U	SocSec:O_1=Stud ID:O_U
Major:O_1=Major:O_U	Field:O_1=Major:O_U
GPA:O_1=Grade:O_U	Grade:O_1=Grade:O_U
Ethnicity:O_1=Ethnicity :O_U	
	Gender:O_2=Gender:O_U
Ethnicity:O_1=Ethnicity :O_U	
Intern :O_1=Internship:O_U	Work-Experience:O_2=Internship:O_U
Placed-In :O_1=Employment-Status:O_U	Has-Status: O_2=Employment-Status:O_U

MAIN FOCUS

Problem Definition

Given a data set D, a hypothesis class H, and a performance criterion P, an algorithm L for learning (from centralized data D) outputs a hypothesis $h \in H$ that optimizes P. In pattern classification applications, h is a classifier (e.g., a decision tree, a support vector machine, etc.). The data D typically consists of a set of training examples. Each training example is an ordered tuple of attribute values, where one of the attributes corresponds to a class label and the remaining attributes represent inputs to the classifier. The goal of learning is to produce a hypothesis that optimizes the performance criterion (e.g., minimizing classification error on the training data) and the complexity of the hypothesis.

In the case of semantically heterogeneous data sources, we assume the existence of:

1. A collection of several related OEDSs $\mathscr{D}_1 = \{ D_1,S_1,O_1\}$,$\mathscr{D}_2 = \{ D_2,S_2,O_2\},\ldots,\mathscr{D}_n = \{D_n,S_n,O_n\}$ for which schemas and ontologies are made explicit, and instances in the data sources are labeled according to some criterion of interest to a user (e.g., employment status).
2. A user view, consisting of a user ontology O_U and a set of mappings $\{\Psi_k\}$ that relate the user ontology O_U to the data source ontologies $O_1,\ldots,O_p$. The user view implicitly specifies a user level of abstraction, corresponding to the leaf nodes of the hierarchies in O_U. The mappings $\{\Psi_k\}$ can be specified manually by a user or semi-automatically derived.
3. A hypothesis class H (e.g., Bayesian classifiers) defined over an *instance space* (implicitly specified by concepts, their properties, and the associated ontologies in the domain of interest) and a performance criterion P (e.g., accuracy on a classification task).

The problem of learning classifiers from a collection of related OEDSs can be simply formulated as follows: under the assumptions (1)-(3), the task of a learner L is to output a hypothesis $h \in H$ that optimizes a criterion P, via the mappings $\{\Psi_k\}$.

As in (Caragea et al., 2005), we say that an algorithm L_S for learning from OEDSs $\mathscr{D}_1,\mathscr{D}_{2\ldots}\mathscr{D}_n$, via the mappings $\{\Psi_k\}$ is *exact* relative to its centralized counterpart L_C, if the hypothesis produced by L_S (federated approach) is identical to that obtained by L_C from the data warehouse D constructed by integrating the data sources $\mathscr{D}_1,\mathscr{D}_{2\ldots}\mathscr{D}_n$, according to the user view, via the same mappings $\{\Psi_k\}$ (data warehouse approach).

The *exactness* criterion defined previously assumes that it is possible, in principle, to create an integrated data warehouse in the centralized setting. However, in practice, the data sources $\mathscr{D}_1,\mathscr{D}_{2\ldots}\mathscr{D}_n$ might impose access constraints Z on a user U. For example, data source constraints might prohibit retrieval of raw data from some data sources (e.g., due to query form access limitations, memory or bandwidth limitations, privacy concerns) while allowing retrieval of answers to statistical queries (e.g., count frequency queries).

Partially Specified Data

Consider the data source ontologies O_1 and O_2 and the user ontology O_U shown in Figure 2. The attribute *Has-Status* in data source D_2 is specified in greater detail (lower level of abstraction) than the corresponding attribute *Placed-In* is in D_1. That is, data source D_2 carries information about the precise status of students after they graduate (specific advanced degree program e.g., *Ph.D.*, *M.S.* that the student has been accepted into, or the type of employment that the student has accepted) whereas data source D_1 makes no distinctions between the types of graduate degrees or types of employment. We say that the *Status* of students in data source D_1 are only *partially specified* (Zhang et al., 2005) with respect to the ontology O_U. In such cases, answering statistical queries from semantically heterogeneous data sources requires the user to supply not only the mapping between the user ontology and the ontologies associated with the data sources but also *additional assumptions of a statistical nature* (e.g., that grad program admits in D_1 and D_2 can be modeled by the same underlying distribution). The validity of the answer returned depends on the validity of the assumptions and the soundness of the procedure that computes the answer based on the supplied assumptions.

Sufficient Statistics Based Solution

Our approach to the problem of learning classifiers from OEDSs is a natural extension of a general strategy for

transforming algorithms for learning classifiers from data in the form of a single flat table (as is customary in the case of a vast majority of standard machine learning algorithms) into algorithms for learning classifiers from a collection of *horizontal* or *vertical* fragments of the data, corresponding to partitions of rows or columns of the flat table, wherein each fragment corresponds to an ontology extended data source.

This strategy, inspired by (Kearns, 1998) involves a decomposition of a learning task into two parts: a *statistics gathering* component, which retrieves the statistics needed by the learner from the distributed data sources, and a *hypothesis refinement* component, which uses the statistics to refine a partially constructed hypothesis (starting with an empty hypothesis) (Caragea et al., 2005).

In the case of learning classifiers from semantically disparate OEDSs, the statistics gathering component has to specify the statistics needed for learning as a *query* against the user view and assemble the answer to this query from OEDSs. This entails: decomposition of a posed query into sub-queries that the individual data sources can answer; translation of the sub-queries to the data source ontologies, via user-specific mappings; query answering from (possibly) partially specified data sources; composition of the partial answers into a final answer to the initial query (Figure 4).

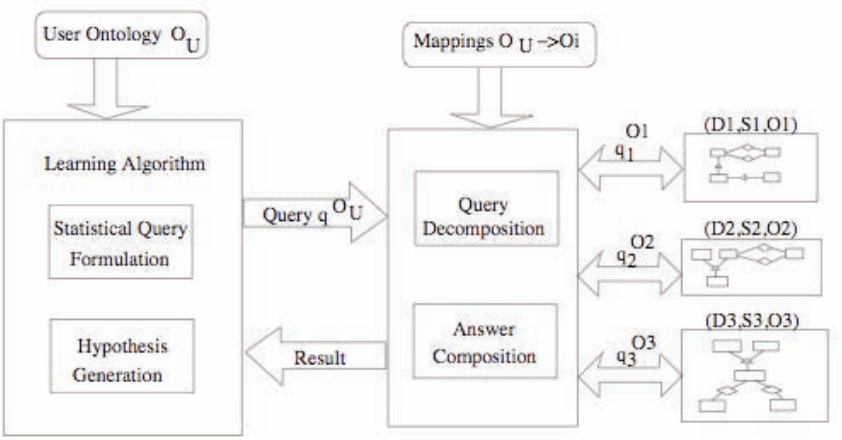
The resulting algorithms for learning from OEDSs are *provably exact* relative to their centralized counterparts, for a family of learning classifiers for which the sufficient statistics take the form of counts of instances satisfying certain constraints on the values of the attributes (e.g., naïve Bayes, decision trees, etc.).

The efficiency of the proposed approach (relative to the centralized setting) depends on the specifics of access constraints and query answering capabilities associated with the individual OEDSs. At present, many data sources on the Web offer query interfaces that can only be used to retrieve small subsets of the data that match a limited set of conditions that can be selected by the user. In order for Web data sources to serve the needs of communities of users interested in building predictive models from the data (e.g., in e-science and other emerging data-rich applications), it would be extremely useful to equip the data sources with statistical query answering capabilities.

FUTURE TRENDS

Some interesting directions for future research include: exploring the effect of using different ontologies and mappings, use of the proposed framework to evaluate mappings, study of the quality of the classifier with respect to the set of mappings used, etc.

Figure 4. Learning classifiers from OEDSs



CONCLUSION

In this chapter, we have precisely formulated the problem of learning classifiers from a collection of several related OEDSs, which make *explicit* (the typically implicit) ontologies associated with the data sources of interest. We have shown how to exploit data sources annotated with relevant meta data in building predictive models (e.g., classifiers) from several related OEDSs, without the need for a centralized data warehouse, while offering strong guarantees of *exactness* of the learned classifiers wrt the centralized traditional relational learning counterparts. User-specific mappings between the user ontology and data source ontologies are used to answer statistical queries that provide the sufficient statistics needed for learning classifiers from OEDSs.

ACKNOWLEDGMENT

This research has been supported by the NSF Grant # 0711396.

REFERENCES

Berners-Lee, T., Hendler, J. and Ora Lassila (2001). The semantic web. *Scientific American*.

Bonatti, P., Deng, Y., and Subrahmanian, V. (2003). An ontology-extended relational algebra. In: *Proceedingsofthe IEEE Conference on Information Integration and Reuse*, pages 192–199. IEEE Press, 2003.

Caragea, D., Zhang, J., Bao, J., Pathak, J. and Honavar, V. (2005). Algorithms and software for collaborative discovery from autonomous, semantically heterogeneous information sources. In *Proceedings of the 16th International Conference on Algorithmic Learning Theory*, volume 3734 of *LNCS*, H. Kargupta and P. Chan. *Advances in Distributed and Parallel Knowledge Discovery*. AAAI/MIT, 2000.

Davidson, S., Crabtree, J., Brunk, B., Schug, J., Tannen, V., Overton, G., Stoeckert, C., (2001). K2/Kleisli and GUS: Experiments in integrated access to genomic data sources. In: *IBM Journal*, Vol. 40, No 2, 2001.

Doan, A. and Halevy, A. (2005). Semantic Integration Research in the Database Community: A Brief Survey, *AI Magazine, Special Issue on Semantic Integration, Spring 2005*.

Eckman, B. (2003). A Practitioner's guide to data management and data integration in Bioinformatics. In: Bioinformatics, Lacroix, Z., and Crithlow, T. (Ed). Palo Alto, CA: Morgan Kaufmann. 2003. pp. 35-74.

Hull, R. (1997). Managing Semantic Heterogeneity in Databases: A Theoretical Perspective. PODS 1997, pages 51-61, Tucson, Arizona.

Levy, A., (1998). The Information Manifold approach to data integration. In: IEEE Intelligent Systems, 13 12-16.

Kalfoglou, Y. and Schorlemmer, M. (2005). Ontology mapping: The state of the art. In *Dagstuhl Seminar Proceedings: Semantic Interoperability and Integration*, Dagstuhl, Germany.

Kargupta, H., and Chan, P. (1999) *Advances in Distributed and Parallel Knowledge Discover*y. Cambridge, MA: MIT Press.

M. Kearns. (1998). Efficient noise-tolerant learning from statistical queries. *Journal of the ACM*, 45(6):983–1006.

Noy, N. (2004). Semantic Integration: A Survey Of Ontology-Based Approaches. *SIGMOD Record, Special Issue on Semantic Integration*, 33(4).

Noy, N. and Stuckenschmidt, H. (2005). Ontology Alignment: An annotated Bibliography. In Y. Kalfoglou, M. Schorlemmer, A. Sheth, S. Staab, and M. Uschold, editors, *Semantic Interoperability and Integration*, Dagstuhl Seminar Proceedings.

Zhang, J., Kang, D-K., Silvescu, A. and Honavar, V. (2005). Learning compact and accurate naive Bayes classifiers from attribute value taxonomies and data. *Knowledge and Information Systems*.

KEY TERMS

Classification Task: A task for which the learner is given experience in the form of labeled examples and it is supposed to learn to classify new unlabeled examples. In a classification task, the output of the learning algorithm is called hypothesis or classifier (e.g., a decision tree, a support vector machine, etc.)

Learning from Semantically Heterogeneous Data Sources: Given a set of related OEDS, a user view (schema, ontology and mappings), a hypothesis class and a performance criterion, the task of the learner is to output a hypothesis that optimizes the performance criterion, via the user mappings. We say that an algorithm for learning from OEDS via a set of mappings is *exact* relative to its centralized counterpart, if the hypothesis it produces is identical to that produced by the centralized learning from the data warehouse constructed by integrating the OEDS, via the same set of mappings.

Learning Task Decomposition: A learning algorithm can be decomposed in two components: (a) an *statistics gathering* component that formulates and sends a statistical query to a data source; and (b) a *hypothesis refinement* component that uses the resulting statistic to modify a partially constructed algorithm output (and further invokes the information extraction component if needed to generate the final algorithm output). In the case of the OESD, the statistics gathering component entails decomposition of a posed query into sub-queries that the individual data sources can answer; translation of the sub-queries to the data source ontologies, via user-specific mappings; query answering from (possibly) partially specified data sources; composition of the partial answers into a final answer to the initial query.

Ontology-Extended Data Sources: An OEDS consists of a data set (representing the instantiation of a schema), a data source schema and a data source ontology. The ontology has two parts, a structure ontology, which defines the semantics of the schema (entities and attributes of entities that appear in data source schema *S*) and a content ontology, which defines the semantics of the data instances (values and relationships between values that the attributes can take in instantiations of the schema).

Sufficient Statistics: A statistic is called a sufficient statistic for a parameter if the statistic captures all the information about the parameter, contained in the data. More generally, a statistic is called a sufficient statistic for learning a hypothesis using a particular learning algorithm applied to a given data set, if there exists an algorithm that takes as input the statistic and outputs the desired hypothesis. A query that returns a statistic is called a statistical query.

User View: A user view of a set of OEDS is specified by a user schema, a user ontology and a set of mapping from the user schema to the data sources schemas, and from the user ontology to the data source ontologies. A data attribute is partially specified wrt a user view, if the the attribute is specified at a higher level of abstraction than the assumed level of abstraction in the user ontology.

Knowledge Discovery in Databases with Diversity of Data Types

QingXiang Wu
University of Ulster at Magee, UK

Martin McGinnity
University of Ulster at Magee, UK

Girijesh Prasad
University of Ulster at Magee, UK

David Bell
Queen's University, UK

INTRODUCTION

Data mining and knowledge discovery aim at finding useful information from typically massive collections of data, and then extracting useful knowledge from the information. To date a large number of approaches have been proposed to find useful information and discover useful knowledge; for example, decision trees, Bayesian belief networks, evidence theory, rough set theory, fuzzy set theory, kNN (k-nearest-neighborhood) classifier, neural networks, and support vector machines. However, these approaches are based on a specific data type. In the real world, an intelligent system often encounters mixed data types, incomplete information (missing values), and imprecise information (fuzzy conditions). In the UCI (University of California – Irvine) Machine Learning Repository, it can be seen that there are many real world data sets with missing values and mixed data types. It is a challenge to enable machine learning or data mining approaches to deal with mixed data types (Ching, 1995; Coppock, 2003) because there are difficulties in finding a measure of similarity between objects with mixed data type attributes. The problem with mixed data types is a long-standing issue faced in data mining. The emerging techniques targeted at this issue can be classified into three classes as follows: (1) Symbolic data mining approaches plus different discretizers (e.g., Dougherty et al., 1995; Wu, 1996; Kurgan et al., 2004; Diday, 2004; Darmont et al., 2006; Wu et al., 2007) for transformation from continuous data to symbolic data; (2) Numerical data mining approaches plus transformation from symbolic data to numerical data (e.g.,, Kasabov, 2003; Darmont et al., 2006; Hadzic et al., 2007); (3) Hybrid of symbolic data mining approaches and numerical data mining approaches (e.g.,, Tung, 2002; Kasabov, 2003; Leng et al., 2005; Wu et al., 2006). Since hybrid approaches have the potential to exploit the advantages from both symbolic data mining and numerical data mining approaches, this chapter, after discassing the merits and shortcomings of current approaches, focuses on applying Self-Organizing Computing Network Model to construct a hybrid system to solve the problems of knowledge discovery from databases with a diversity of data types. Future trends for data mining on mixed type data are then discussed. Finally a conclusion is presented.

BACKGROUND

Each approach for data mining or knowledge discovery has its own merits and shortcomings. For example, EFNN (Evolving Fuzzy Neural Network based on Tokagi-Sgeno fuzzy rules) (Kasabov, 2003; Takagi and Sugeno, 1985), SOFNN (Leng et al., 2005; Kasabov, 2003; Tung, 2002), dynamic fuzzy neural networks, kNN, neural networks, and support vector machines, are good at dealing with continuous valued data. For example, the EFNN (Kasabov, 2003) was applied to deal with benchmark data sets--the gas furnace times series data and the Mackey-Glass time series data (Jang, 1993). High accuracies were reached in the predictive results. The errors were very small i.e. 0.156 for the

Gas-furnace case and 0.039 for the Mackey-Glass case. However, they cannot be directly applied to symbolic data or to a data set with missing values. Symbolic AI techniques (Quinlan, 1986, Quinlan, 1996, Wu et al., 2005) are good at dealing with symbolic data and data sets with missing values. In order to discover knowledge from a database with mixed-type data, traditional symbolic AI approaches always transform continuous valued data to symbolic data. For example, the temperature is a continuous data, but it can be transformed to symbolic data 'cool', 'warm', 'hot', etc. This is a typical transformation of one dimension of continuous data, which is called *discretization*. The transformation for two or more dimensions of continuous data such as pictures or videos can be regarded as object recognition or content extraction. However, information about distance and neighborhood in continuous valued data is ignored if the discretized values are treated as symbolic values in symbolic AI techniques.

On the other hand, symbolic data can be transformed to numerical data using some encoding scheme. This can be done by statistic, rough sets or fuzzy membership functions. For example, 'high', 'mid', and 'low' can be sorted in a sequence and can be represented by fuzzy member functions. However, it is difficult to encode symbols without an explicit sequence, e.g., symbolic values for furniture: 'bed', 'chair', 'bench', 'desk' and 'table'. If the symbols have to be sorted out in a sequence, some additional information is required. For example, they can be sorted by their size or price if the information of price or size are known. Therefore, correct data transformation plays a very important role in data mining or machine learning.

MAIN FOCUS

The Self-Organizing Computing Network Model (Wu et al., 2006) provides a means to combine the transformations and data mining/knowledge discovery approaches to extract useful knowledge from databases with a diversity of data types, and the knowledge is represented in the form of a computing network. The model is designed using a hybrid of symbolic and numerical approaches. Through an analysis of which data type is suitable to which data mining or machine learning approach, data are reclassified into two new classes -- *order dependent attribute* and *order independent attribute*. Then concepts of fuzzy space, statistical learning, neural networks and traditional AI technologies are integrated to the network model to represent knowledge and self-adapt to an instance information system for decision making.

Proper Data Type Transformation

Usually, data can be categorized in two types, i.e. numerical data and symbolic data. From a data mining or machine learning point of view, attribute values can be separated into two classes. If the values of an attribute can be sorted out in a sequence and a distance between two values is significant to data mining or machine learning, the attribute is called an *order dependent attribute*. Numerical attribute can be separated into two kinds of attributes, i.e. a continuous valued attribute and an encoding numerical attribute. A continuous valued attribute is an *order dependent attribute* because a distance between two values can be used to describe neighbors or similarities in data mining or machine learning algorithms. Some encoding numerical attributes are an *order dependent attribute* such as grade numbers 1 to 5 for student courses. Some encoding numerical data such as product identification numbers are not an *order dependent attribute*. There is a distance between two values, but the distance is not significant to data mining or machine learning algorithms. We cannot say product No.1 and product No.2 certainly have similarity or can be regarded as neighbors. If this distance is used in data mining or machine learning algorithms, the results will be degraded. If the values of an attribute cannot be sorted out in a sequence or a distance between two values is not significant to data mining or machine learning, the attribute is called an *order independent attribute*. For example, some symbolic attributes are an *order independent attribute* in which there is neither definition of a distance between two symbolic values nor definition of value sequences or neighborhoods. However, there are some symbolic data with an explicit sequence; for example, 'high', 'mid', and 'low'. These symbolic values are suitable for transfer to numerical data so that value sequence and neighborhood can be used by data mining or machine learning algorithms. Therefore, two attribute channels are designed in the input layer of the Self-Organizing Computing Network Model to lead an attribute with a given data type to a suitable data mining approach. The first channel is called an *order dependent attribute* channel. The data mining or machine learning approaches, which can take

advantages from value sequence or neighborhoods, are designed in this channel. The second channel is called an *order independent attribute* channel. The data mining or machine learning approaches, which do not need the information of value sequence and neighborhood, are designed in this channel.

Structure of Self-Organizing Computing Network

The self-organizing computing network as defined in (Wu et al., 2006) is constructed by means of three classes of computing cells: input cells, hidden cells and output cells, as shown in Figure 1.

In general, computing cells can have different computing mechanisms; for example, neuron models, mathematical functions, computer programs, or processor units. For a self-organizing computing network, input cells are defined as a set of encoders that transform different data values to the internal representation of the network. An input cell corresponds to an attribute in an instance information system (Wu et al., 2005). Therefore, the number of input cells is equal to the number of attributes in the instance information system. Each input cell is connected to hidden cells by means of connection *transfer functions* in matrix T_{IxH} instead of single weights.

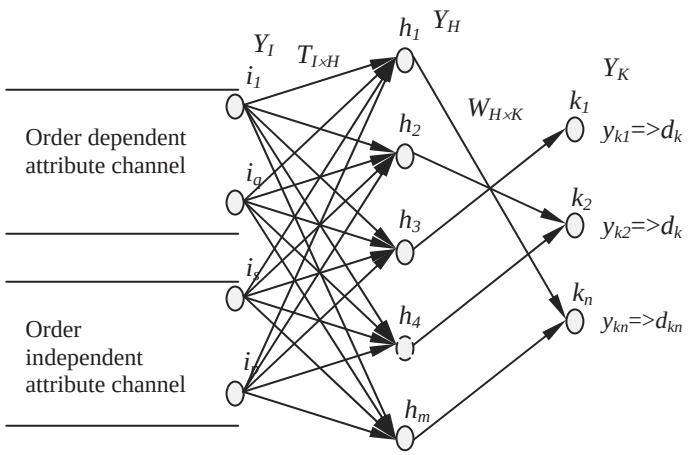
$$T_{I\times H}=\begin{Bmatrix} T^{C}_{i1,h1} & T^{C}_{i1,h2} & \cdots & T^{C}_{i1,h_m} \\ \cdots & \cdots & \cdots & \cdots \\ T^{C}_{iq,h1} & T^{C}_{iq,h2} & \cdots & T^{C}_{iq,h_m} \\ T^{S}_{is,h1} & T^{S}_{is,h1} & \cdots & T^{S}_{is,h_m} \\ \cdots & \cdots & \cdots & \cdots \\ T^{S}_{ip,h1} & T^{S}_{ip,h1} & \cdots & T^{S}_{is,h_m} \end{Bmatrix} \quad (1)$$

Where $T^{c}_{i1,h1}$ represents a transfer function for connection between continuous valued input cell i_1 and hidden cell h_1, $T^{S}_{is,h1}$ represents a transfer function for connection between symbolic input cell i_s and hidden cell h_1. Hidden computing cells are connected to output cells according to the decision values of known instances in the instance information system by a weight matrix W_{HxK}.

$$W_{H\times K}=\begin{Bmatrix} w_{h1,k1} & w_{h1,k2} & w_{h1,k_n} \\ \cdots & \cdots & \cdots \\ w_{h_m,k1} & w_{h_m,k2} & w_{h_m,k_n} \end{Bmatrix} \quad (2)$$

The number of output cells is equal to the number of decisions in the instance information system. The weights are real numbers in the range [0, 1]. A weight in the weight matrix can be regarded as a *decision support degree* for a hidden cell. For example, weight $w_{h,k}$=1 means that hidden cell *h* supports the decision

Figure 1. Architecture of a Self-Organizing Computing Network. $x_1,\ldots x_q$ represent q inputs for order dependent attribute channel. $x_s,\ldots x_p$ represent p-s+1 inputs for order independent attribute channel. $Y_I=\{y_{i1},y_{i2},\ldots y_q,y_s,\ldots y_p\}$ represents a set of outputs of first layer. $I=\{i_1,\ldots i_p\}$. $H=\{h_1,\ldots h_m\}$. $K=\{k_1,\ldots k_m\}$. $Y_H=\{y_{h1},\ldots y_{hm}\}$ represents a set of outputs of hidden layer. $Y_K=\{y_{k1},\ldots y_{kn}\}$ represents a set of outputs of output layer. T_{IxH} represents transfer function matrix. $W_{H\times K}$ represents weight matrix. $V_d=\{d_{k1},\ldots d_{kn}\}$ represents a set of decisions.



corresponding to output cell k. Weight $w_{h,k}$=0 means that hidden cell h does not support the decision corresponding to output cell k.

Internal Information Representation

Through two channels in the Self-Organizing Computing Network Model, input data can be transformed to an internal information representation, which is represented by an interval of integer numbers such as [1, 50], or [1,100] because an integer number can be regarded as not only discrete continuous data but also code of symbolic data. Mixed-type input data are transferred to internal representation by the input cells.

$$y_i = (x_i - x_{i(\min)})s_i,\ s_i = s_w / (x_{i(\max)} - x_{i(\min)}),\ \text{for } i \in I_c\ , \tag{3}$$

For order independent symbolic data, we have

$$y_i = Numerical\ Code(x_i)\ ,\ \text{for } i \in I_s \tag{4}$$

where $I_c = \{i_1, \ldots i_q\}$, $I_s = \{i_s, \ldots i_p\}$, and s_w is the encoding range. In order to encode symbolic data, s_w is set to an integer for example 50 or 100.

A cell in the hidden layer corresponds to a fuzzy rule. The connections from the hidden cell to input cells are regarded as fuzzy conditions. Each input cell in the *order dependent attribute* channel is connected to the hidden layer by a *self-adaptive sense-function* as follows:

$$T_{i,h}^{C}(y_i) = \begin{cases} e^{-\frac{(y_i - y_{L(i,h)})^2}{\delta_{i,h}^2}} & y_i < y_{L(i,h)} \\ 1 & y_{L(i,h)} \le y_i \le y_{U(i,h)} \\ e^{-\frac{(y_i - y_{U(i,h)})^2}{\delta_{i,h}^2}} & y_i > y_{U(i,h)} \end{cases} \tag{5}$$

where y_i is a value from input cell i, $y_{L(i,h)}$ is the lower fuzzy edge, $y_{U(i,h)}$ is the upper fuzzy edge, and $\delta_{i,h}$ is a fuzzy degree for both edges. Constants $y_{L(i,h),}$ $y_{U(i,h),}$ and $\delta_{i,h.}$ can be adjusted by the learning algorithm.

Each input cell in the *order independent attribute* channel is connected to the hidden layer by a *statistical learning function* (Wu et al., 2005) as follows. Suppose that one hidden cell supports only one decision $d_h \in V_d$.

$$T_{i,h}^{S}(y_i) = Sup(y_i \mid d_h) = \frac{N_{xi,yi,h} + \beta\ |U|}{N_{xi,dh} + \beta\ |U|\,|V_{xi}|_{\mathrm{f}}} \tag{6}$$

where U is the set of all instances in an instance information system, $|U|$ is the total number of instances, $N_{xi,yi,h}$ is the number of attribute values y_i that supports decision d_h, $N_{xi,dh}$ is the number of instances with decision d_h in U, V_{xi} is the domain of attribute x_i, $|V_{xi}|$ is the number of symbolic values in V_{xi}, and β is a small number with typical value 0.02. Through the transformation of the two types of connection functions, the input data is transferred to a *conditional matched-degree*. A hidden cell can integrate *conditional matched-degrees* from all the input cells, and then output an overall *matched-degree*. It is expressed as follows:

$$\begin{aligned} y_h(Y_I) &= y_h(Y_{I_c})\, y_h(Y_{I_s}) \\ &= [\prod_{i \in I_c} T_{i,h}^{C}(y_i)]^{\frac{1}{q}} [\prod_{i \in I_s} T_{i,h}^{S}(y_i)]^{\frac{1}{p-q}} \qquad \text{for } h \in H \end{aligned} \tag{7}$$

Connection weights between hidden layer and output layer can be regarded as *support degree* for a hidden cell to support a decision corresponding to an output cell. Connection weights can be trained by training sets. An output value of an output cell in output layer is defined as *belief* for a specific decision.

Self-Organizing

Corresponding to the multiple inputs, each hidden cell has a specific *fuzzy sub-superspace* determined by the *self-adaptive sense-function*s and *statistical learning function* in Equation . The parameters lower fuzzy edge $y_{L(i,h)}$ and high fuzzy edge $y_{U(i,h)}$ for the *fuzzy sub-superspace* are self-organized by adjusting sense-range so that $y_{L(i,h)} = L_{i,\,hj\ (\min)}$ and $y_{U(i,h)} = U_{i,\,hj\ (\min)}$, and the hidden cell can adapt to the decision $d \in V_d$. The algorithm for this adjusting is as follows:

$$L_{i,hj\,(\min)} = \min\{L_{i,hj} : y_{hj}(u) < \theta \ \ for\ u \notin TR_d\ ), \text{for all } i \in \{1, 2, \ldots, p\}$$

$$U_{i,hj(\max)} = \max\{U_{i,hj} : y_{hj}(u) < \theta \ \ for\ u \notin TR_d\ ), \text{for all } i \in \{1, 2, \ldots, p\}$$

where u is an instance with attributes $(x_1, \ldots x_p)$, a decision value d, TR_d is a set of training instances, and θ

is a threshold value. The weights between the hidden layer and output layer are also trained using instances in training sets. After training, each hidden cell has a specific *response sub-superspace*. In order to make sure that the algorithm converges to an optimal *response sub-superspace*, the order for attributes to adjust the sense function parameters can be determined by important degree (Wu et al., 2005). A tolerance can be used to stop the adjusting (Wu et al., 2006).

Decision Making

A Self-Organizing Computing Network can be created using a training set from an instance information system. After a specific Self-Organizing Computing Network is obtained, the network can be applied to make decisions. If a set of mixed-type values presents to the network, the network take in the data from two channels, i.e. the order dependent attribute channel and order independent attribute channel. The data are transferred to internal representation by the first layer cells, and then transferred to condition-matched degree by the sense-functions. The hidden cells integrate the condition-matched degrees. The output cells will give the belief distribution over the decision space.

$$y_{d(\max)} = \max_{h \in H}(w_{h,d}\, y_h) \text{ for } d \in \boldsymbol{V}_d \qquad (8)$$

The maximal belief can be used to make a final decision.

$$d_M = \arg\max_{d \in V_d}(y_{d(\max)}) \qquad (9)$$

This approach has been applied to data sets from the UCI Machine Learning Repository. The decision accuracies obtained by the proposed approach were better than other approaches (Wu et al., 2006).

FUTURE TRENDS

As mentioned in the background section, a single data mining approach is very unlikely to solve a mixed type data problem. Hybrid approaches, multiple classifier approaches, and biologically inspired approaches are the trends to improve the solutions of this problem. The Self-Organizing Computing Network Model is a general model. Based on the model, the connection transfer function, computing mechanism of the cells and training algorithms can be changed and it is possible to find better components for the network model. In current researches, the defined sense-function in equation is applied to the computing network model. Using this function and the proposed sense function adjusting algorithm, the fuzzy sub-superspace is expanded by grids. It is possible to apply other functions to enable the fuzzy sub-superspace to be expanded elliptically. For order independent inputs, the modified Naïve Bayes approach (Wu et al., 2005) has been integrated in the Computing Network Model. This also can be tried with decision tree and rough set approaches, etc. The computing mechanism of hidden cells is a multiplier in the proposed network. However, other computing mechanisms can replace the multiplier. These are topics for further study.

CONCLUSION

The Self-Organizing Computing Network provides a means to create a hybrid of symbolic AI techniques and computational intelligence approaches. In this approach, the input layer is designed as a set of data converters. Each hidden cell responds to a fuzzy sub-superspace within the input superspace. The fuzzy sub-superspace of a hidden cell adapts to the training set according to the self-organizing algorithms. Based on the definitions for the computing cell and its connections, each output value can be explained using symbolic AI concepts such as belief, support degree, match degree, or probability. Using this network, instance information systems with mixed data types can be handled directly. Note that preparation of input data is included in the self-organizing algorithm. Therefore, this approach has neither directly given results for data reduction and aggregation, nor for structural and semantic heterogeneity. It may be possible to extend the cells in the hidden layer to retrieve these. This is a topic for further study.

REFERENCES

Ching, J. Y., Wong, A.K.C., Chan, K.C.C. (1995). Class-dependent discretization for inductive learning from continuous and mixed-mode data, IEEE Transactions on Pattern Analysis and Machine Intelligence, 17(7)641-651.

Coppock, S., Mazlack, L. (2003). Soft multi-modal data fusion, The 12th IEEE International Conference on Fuzzy Systems, 1, 25-28.

Darmont, J., Boussaïd, O., Eds. (2006) *Processing and Managing Complex Data for Decision Support*, Idea Group Publishing.

Diday, E., Vrac, M. (2005). Mixture Decomposition of Distributions by Copulas In the Symbolic Data Analysis Framework. Journal of Discrete Applied Mathematics (DAM). 147(1)27-41.

Diday, E., Murty, N., (2005). Symbolic Data Clustering" in Encyclopedia of Data Warehousing and Mining . John Wong editor . Idea Group Reference Publisher.

Dougherty, J., Kohavi, R., Sahami, M., (1995). Supervised and Unsupervised Discretization of Continuous Features, Proc. of Int'l Conf. Machine Learning, 194-202.

Hadzic, F., Dillon,T. S. (2007). CSOM for Mixed Data Types, Advances in Neural Networks – ISNN 2007, LNCS, Springer Berlin, 4492, 965-978.

Jang, J.S. R.(1993). Adaptive-Network-Based Fuzzy Inference Systems, IEEE Trans. on System, Man, and Cybernetics, 23(3)665-685.

Kasabov, N. (2003). Evolving Connectionist Systems-Methods and Applications, Brain Study and intelligent Machines, Springer – Verlag London Limited.

Kurgan, L. A. and Cios, K. J. (2004). CAIM Discretization Algorithm, IEEE Transactions on Knowledge and Data Engineering, 16(2)145-153.

Leng, G., McGinnity, T. M., Prasad, G. (2005). An approach for on-line extraction of fuzzy rules using a self-organising fuzzy neural network, Fuzzy Sets and Systems, Elsevier, 150(2) 211-243.

Quinlan, J. R. (1986). Induction of Decision Trees, Machine Learning, 1(1) 81 – 106,.

Quinlan, J. R. (1996). Improved Use of Continuous Attributes in C4.5, Journal of Artificial Intelligent Research, Morgan Kaufmann Publishers, 4, 77-90.

Takagi, T. and Sugeno, M., (1985). Fuzzy identification of systems and its applications to modeling and control, IEEE Trans. Syst., Man, Cybern., 15(1)116–132.

Tung, W. L., Quek, C. (2002). GenSoFNN: a generic self-organizing fuzzy neural network," IEEE Transactions on Neural Networks, 13(5)1075 – 1086.

Wu, Q. X., Bell, D. A., and McGinnity, T. M. (2005). Multi-knowledge for decision making, International Journal of Knowledge and Information Systems, Springer-Verlag, 7(2) 246– 266.

Wu, Q. X., Bell, D. A., Prasad, G. , McGinnity, T. M. (2007). A Distribution-Index-Based Discretizer for Decision-Making with Symbolic AI Approaches, IEEE Transaction on Knowledge and Data Engineering, 19(1)17-28.

Wu, Q. X., McGinnity, T. M., Bell, D. A., Prasad, G. (2006). A Self-Organising Computing Network for Decision-Making in Data Sets with Diversity of Data Types, IEEE Transaction on Knowledge and Data Engineering, 18(7) 941-953.

Wu, X. (1996). A Bayesian Discretizer for Real-Valued Attributes," The Computer Journal, 39(8)688-691.

KEY TERMS

Condition Match Degree: A value in [0,1] represents how much a given set of condition attribute values match the Response Sub-Superspace.

Fuzzy Sub-Superspace: A sub space in a multiple dimension input space is determined by a set of sense-functions.

Order Dependent Attribute: Values of an attribute can be sorted out in a sequence and a distance between two values is significant to data mining or machine learning.

Order Independent Attribute: There are no sequence definition and distance definition for values of an attribute.

Response Sub-Superspace: A fuzzy sub-superspace is determined by a training set.

Self-Organizing Computing Network Model: A network is defined by flexible computing cells and connection functions for hybrid of different data mining or machine learning approaches.

Sense-Function: A fuzzy member function is defined to connect cells between the first layer cells and hidden layer cells in the Self-Organizing Computing Network.

Support Degree: As each hidden cell represents a fuzzy rule in a Self-Organizing Computing Network, the output of a hidden cell is a condition match-degree. The product of this condition match-degree and the connection weight is regarded as a Support Degree for the hidden cell to corresponding output cell.

Learning Bayesian Networks

Marco F. Ramoni
Harvard Medical School, USA

Paola Sebastiani
Boston University School of Public Health, USA

INTRODUCTION

Born at the intersection of artificial intelligence, statistics, and probability, Bayesian networks (Pearl, 1988) are a representation formalism at the cutting edge of knowledge discovery and data mining (Heckerman, 1997). Bayesian networks belong to a more general class of models called *probabilistic graphical models* (Whittaker, 1990; Lauritzen, 1996) that arise from the combination of graph theory and probability theory, and their success rests on their ability to handle complex probabilistic models by decomposing them into smaller, amenable components. A probabilistic graphical model is defined by a graph, where nodes represent stochastic variables and arcs represent dependencies among such variables. These arcs are annotated by probability distribution shaping the interaction between the linked variables. A probabilistic graphical model is called a Bayesian network, when the graph connecting its variables is a directed acyclic graph (DAG). This graph represents conditional independence assumptions that are used to factorize the joint probability distribution of the network variables, thus making the process of learning from a large database amenable to computations. A Bayesian network induced from data can be used to investigate distant relationships between variables, as well as making prediction and explanation, by computing the conditional probability distribution of one variable, given the values of some others.

BACKGROUND

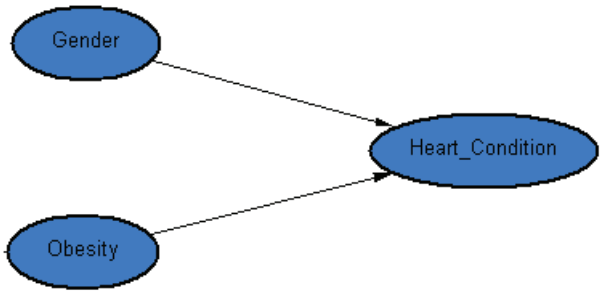
The origins of Bayesian networks can be traced back as far as the early decades of the 20th century, when Sewell Wright developed path analysis to aid the study of genetic inheritance (Wright, 1923, 1934). In their current form, Bayesian networks were introduced in the early 1980s as a knowledge representation formalism to encode and use the information acquired from human experts in automated reasoning systems in order to perform diagnostic, predictive, and explanatory tasks (Charniak, 1991; Pearl, 1986, 1988). Their intuitive graphical nature and their principled probabilistic foundations were very attractive features to acquire and represent information burdened by uncertainty. The development of amenable algorithms to propagate probabilistic information through the graph (Lauritzen, 1988; Pearl, 1988) put Bayesian networks at the forefront of artificial intelligence research. Around the same time, the machine-learning community came to the realization that the sound probabilistic nature of Bayesian networks provided straightforward ways to learn them from data. As Bayesian networks encode assumptions of conditional independence, the first machine-learning approaches to Bayesian networks consisted of searching for conditional independence structures in the data and encoding them as a Bayesian network (Glymour, 1987; Pearl, 1988). Shortly thereafter, Cooper and Herskovitz (1992) introduced a Bayesian method that was further refined by Heckerman, et al. (1995) to learn Bayesian networks from data.

These results spurred the interest of the data-mining and knowledge-discovery community in the unique features of Bayesian networks (Heckerman, 1997); that is, a highly symbolic formalism, originally developed to be used and understood by humans, well-grounded on the sound foundations of statistics and probability theory, able to capture complex interaction mechanisms and to perform prediction and classification.

MAIN THRUST

A Bayesian network is a graph, where nodes represent stochastic variables and (arrowhead) arcs represent

Figure 1.



dependencies among these variables. In the simplest case, variables are discrete, and each variable can take a finite set of values.

Representation

Suppose we want to represent the variable *gender*. The variable gender may take two possible values: male and female. The assignment of a value to a variable is called the *state of the variable*. So, the variable gender has two states: Gender = Male and Gender = Female. The graphical structure of a Bayesian network looks like this:

The network represents the notion that obesity and gender affect the heart condition of a patient. The variable obesity can take three values: yes, borderline and no. The variable heart condition has two states: true and false. In this representation, the node heart condition is said to be a *child* of the nodes gender and obesity, which, in turn, are the *parents* of heart condition.

The variables used in a Bayesian networks are stochastic, meaning that the assignment of a value to a variable is represented by a probability distribution. For instance, if we do not know for sure the gender of a patient, we may want to encode the information so that we have better chances of having a female patient rather than a male one. This guess, for instance, could be based on statistical considerations of a particular population, but this may not be our unique source of information. So, for the sake of this example, let's say that there is an 80% chance of being female and a 20% chance of being male. Similarly, we can encode that the incidence of obesity is 10%, and 20% are borderline cases. The following set of distributions tries to encode the fact that obesity increases the cardiac risk of a patient, but this effect is more significant in men than women:

The dependency is modeled by a set of probability distributions, one for each combination of states of the variables gender and obesity, called the parent variables of heart condition.

Figure 2.

Heart_Condition

Obesity	Gender	True	False
Yes	Male	0.800	0.200
Yes	Female	0.700	0.300
Borderline	Male	0.750	0.250
Borderline	Female	0.600	0.400
No	Male	0.200	0.800
No	Female	0.100	0.900

Learning

Learning a Bayesian network from data consists of the induction of its two different components: (1) the graphical structure of conditional dependencies (model selection) and (2) the conditional distributions quantifying the dependency structure (parameter estimation).

There are two main approaches to learning Bayesian networks from data. The first approach, known as constraint-based approach, is based on conditional independence tests. As the network encodes assumptions of conditional independence, along this approach we need to identify conditional independence constraints in the data by testing and then encoding them into a Bayesian network (Glymour, 1987; Pearl, 1988; Whittaker, 1990).

The second approach is Bayesian (Cooper & Herskovitz, 1992; Heckerman et al., 1995) and regards model selection as an hypothesis testing problem. In this approach, we suppose to have a set M= $\{M_0, M_1, ..., M_g\}$ of Bayesian networks for the random variables $Y_1, ..., Y_v$,, and each Bayesian network represents an hypothesis on the dependency structure relating these variables. Then, we choose one Bayesian network after observing a sample of data D = $\{y_{1k}, ..., y_{vk}\}$, for $k = 1, \ldots, n$. If $p(M_h)$ is the prior probability of model M_h, a Bayesian solution to the model selection problem consists of choosing the network with maximum posterior probability:

$p(M_h|D) \propto p(M_h)p(D|M_h)$.

The quantity $p(M_h|D)$ is the marginal likelihood, and its computation requires the specification of a parameterization of each model M_h and the elicitation of a prior distribution for model parameters. When all variables are discrete or all variables are continuous, follow Gaussian distributions, and the dependencies are linear and the marginal likelihood factorizes into the product of marginal likelihoods of each node and its parents. An important property of this likelihood modularity is that in the comparison of models that differ only for the parent structure of a variable Y_i, only the local marginal likelihood matters. Thus, the comparison of two local network structures that specify different parents for Y_i can be done simply by evaluating the product of the local Bayes factor $BF_{h,k} = p(D|M_{hi}) / p(D|M_{ki})$, and the ratio $p(M_{hi})/ p(M_{ki})$, to compute the posterior odds of one model vs. the other as $p(M_{hi}|D) / p(M_{ki}|D)$.

In this way, we can learn a model locally by maximizing the marginal likelihood node by node. Still, the space of the possible sets of parents for each variable grows exponentially with the number of parents involved, but successful heuristic search procedures (both deterministic and stochastic) exist to render the task more amenable (Cooper & Herskovitz, 1992; Singh & Larranaga, 1996; Valtorta, 1995).

Once the structure has been learned from a dataset, we still need to estimate the conditional probability distributions associated to each dependency in order to turn the graphical model into a Bayesian network. This process, called *parameter estimation*, takes a graphical structure and estimates the conditional probability distributions of each parent-child combination. When all the parent variables are discrete, we need to compute the conditional probability distribution of the child variable, given each combination of states of its parent variables. These conditional distributions can be estimated either as relative frequencies of cases or, in a Bayesian fashion, by using these relative frequencies to update some, possibly uniform, prior distribution. A more detailed description of these estimation procedures for both discrete and continuous cases is available in Ramoni and Sebastiani (2003).

Prediction and Classification

Once a Bayesian network has been defined, either by hand or by an automated discovery process from data, it can be used to reason about new problems for prediction, diagnosis, and classification. Bayes' theorem is at the heart of the propagation process.

One of the most useful properties of a Bayesian network is the ability to propagate evidence irrespective of the position of a node in the network, contrary to standard classification methods. In a typical classification system, for instance, the variable to predict (i.e., the class) must be chosen in advance before learning the classifier. Information about single individuals then will be entered, and the classifier will predict the class (and only the class) of these individuals. In a Bayesian network, on the other hand, the information about a single individual will be propagated in any direction in the network so that the variable(s) to predict must not be chosen in advance.

Although the problem of propagating probabilistic information in Bayesian networks is known to be, in the general case, NP-complete (Cooper, 1990), several scalable algorithms exist to perform this task in networks with hundreds of nodes (Castillo, et al., 1996; Cowell et al., 1999; Pearl, 1988). Some of these propagation algorithms have been extended, with some restriction or approximations, to networks containing continuous variables (Cowell et al., 1999).

FUTURE TRENDS

The technical challenges of current research in Bayesian networks are focused mostly on overcoming their current limitations. Established methods to learn Bayesian networks from data work under the assumption that each variable is either discrete or normally distributed around a mean that linearly depends on its parent variables. The latter networks are termed *linear Gaussian* networks, which still enjoy the decomposability properties of the marginal likelihood. Imposing the assumption that continuous variables follow linear Gaussian distributions and that discrete variables only can be parent nodes in the network but cannot be children of any continuous node, leads to a closed-form solution for the computation of the marginal likelihood (Lauritzen, 1992). The second technical challenge is

the identification of sound methods to handle incomplete information, either in the form of missing data (Sebastiani & Ramoni, 2001) or completely unobserved variables (Binder et al., 1997). A third important area of development is the extension of Bayesian networks to represent dynamic processes (Ghahramani, 1998) and to decode control mechanisms.

The most fundamental challenge of Bayesian networks today, however, is the full deployment of their potential in groundbreaking applications and their establishment as a routine analytical technique in science and engineering. Bayesian networks are becoming increasingly popular in various fields of genomic and computational biology—from gene expression analysis (Friedman, 2004) to proteimics (Jansen et al., 2003) and genetic analysis (Lauritzen & Sheehan, 2004)—but they are still far from being a received approach in these areas. Still, these areas of application hold the promise of turning Bayesian networks into a common tool of statistical data analysis.

CONCLUSION

Bayesian networks are a representation formalism born at the intersection of statistics and artificial intelligence. Thanks to their solid statistical foundations, they have been turned successfully into a powerful data-mining and knowledge-discovery tool that is able to uncover complex models of interactions from large databases. Their high symbolic nature makes them easily understandable to human operators. Contrary to standard classification methods, Bayesian networks do not require the preliminary identification of an outcome variable of interest, but they are able to draw probabilistic inferences on any variable in the database. Notwithstanding these attractive properties and the continuous interest of the data-mining and knowledge-discovery community, Bayesian networks still are not playing a routine role in the practice of science and engineering.

REFERENCES

Binder, J. et al. (1997). Adaptive probabilistic networks with hidden variables. *Mach Learn, 29*(2-3), 213-244.

Castillo, E. et al. (1996). *Expert systems and probabiistic network models*. New York: Springer.

Charniak, E. (1991). Bayesian networks without tears. *AI Magazine, 12*(8), 50-63.

Cooper, G.F. (1990). The computational complexity of probabilistic inference using Bayesian belief networks. *Artif Intell, 42*(2-3), 393-405.

Cooper, G.F., & Herskovitz, G.F. (1992). A Bayesian method for the induction of probabilistic networks from data. *Mach Learn, 9*, 309-347.

Cowell, R.G., et al. (1999). *Probabilistic networks and expert systems*. New York: Springer.

Friedman, N. (2004). Inferring cellular networks using probabilistic graphical models. *Science, 303*, 799-805.

Ghahramani, Z. (1998). Learning dynamic Bayesian networks. In C.L. Giles, & M. Gori (Eds.), *Adaptive processing of sequences and data structures* (pp. 168-197). New York: Springer.

Glymour, C., Scheines, R., Spirtes, P., & Kelly, K. (1987). *Discovering causal structure: Artificial intelligence, philosophy of science, and statistical modeling*. San Diego, CA: Academic Press.

Heckerman, D. (1997). Bayesian networks for data mining. *Data Mining and Knowledge Discovery, 1*(1), 79-119.

Heckerman, D. et al. (1995). Learning Bayesian networks: The combinations of knowledge and statistical data. *Mach Learn, 20*, 197-243.

Jansen, R. et al. (2003). A Bayesian networks approach for predicting protein-protein interactions from genomic data. *Science, 302*, 449-453.

Larranaga, P., Kuijpers, C., Murga, R., & Yurramendi, Y. (1996). Learning Bayesian network structures by searching for the best ordering with genetic algorithms. *IEEE T Syst Man Cyb, 26*, 487-493.

Lauritzen, S.L. (1992). Propagation of probabilities, means and variances in mixed graphical association models. *J Amer Statist Assoc, 87*, 1098-108.

Lauritzen, S.L. (1996). *Graphical models*. Oxford: Clarendon Press.

Lauritzen, S.L. (1988). Local computations with probabilities on graphical structures and their application to expert systems (with discussion). *J Roy Stat Soc B Met, 50*, 157-224.

Lauritzen, S.L., & Sheehan, N.A. (2004). Graphical models for genetic analysis. *Statist Sci, 18*(4), 489-514.

Pearl, J. (1986). Fusion, propagation, and structuring in belief networks. *Artif. Intell., 29*(3), 241-288.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. San Francisco: Morgan Kaufmann.

Ramoni, M., & Sebastiani, P. (2003). Bayesian methods. In M.B. Hand (Ed.), *Intelligent data analysis: An introduction* (pp. 128-166). New York: Springer.

Sebastiani, P., & Ramoni, M. (2001). Bayesian selection of decomposable models with incomplete data. *J Am Stat Assoc, 96*(456), 1375-1386.

Singh, M., & Valtorta, M. (1995). Construction of Bayesian network structures from data: A brief survey and an efficient algorithm. *Int J Approx Reason, 12*, 111-131.

Whittaker, J. (1990). *Graphical models in applied multivariate statistics*. New York: John Wiley & Sons.

Wright, S. (1923). The theory of path coefficients: A reply to Niles' criticisms. *Genetics, 8*, 239-255.

Wright, S. (1934). The method of path coefficients. *Ann Math Statist, 5*, 161-215.

KEY TERMS

Bayes Factor: Ratio between the probability of the observed data under one hypothesis divided by its probability under an alternative hypothesis.

Conditional Independence: Let X, Y, and Z be three sets of random variables; then X and Y are said to be conditionally independent given Z, if and only if $p(x|z,y)=p(x|z)$ for all possible values x, y, and z of X, Y, and Z.

Directed Acyclic Graph (DAG): A graph with directed arcs containing no cycles; in this type of graph, for any node, there is no directed path returning to it.

Probabilistic Graphical Model: A graph with nodes representing stochastic variables annotated by probability distributions and representing assumptions of conditional independence among its variables.

Statistical Independence: Let X and Y be two disjoint sets of random variables; then X is said to be independent of Y, if and only if $p(x)=p(x|y)$ for all possible values x and y of X and Y.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 674-677, copyright 2005 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

Learning Exceptions to Refine a Domain Expertise

Rallou Thomopoulos
INRA/LIRMM, France

INTRODUCTION

This chapter deals with the problem of the cooperation of heterogeneous knowledge for the construction of a domain expertise, and more specifically for the discovery of new unexpected knowledge. Two kinds of knowledge are taken into account:

- Expert statements. They constitute generic knowledge which rises from the experience of domain experts and describes commonly admitted mechanisms that govern the domain. This knowledge is represented as conceptual graph rules, which has the advantage to combine a logic-based formalism and an equivalent graphical representation, essential for non-specialist users (Bos, 1997).
- Experimental data, given by international literature of the domain. They are represented in the relational model. These numerous data describe in detail, in a quantitative way, experiments that were carried out to deepen the knowledge of the domain, and the obtained results. These results may confirm the knowledge provided by the expert statements – or not.

The cooperation of both kinds of knowledge aims, firstly, at testing the validity of the expert statements within the experimental data, secondly, at discovering refinements of the expert statements to consolidate the domain expertise.

Two major differences between the two formalisms are the following. Firstly, the conceptual graphs represent knowledge at a more generic level than the relational data. Secondly, the conceptual graph model includes an ontological part (hierarchized vocabulary that constitutes the support of the model), contrary to the relational model.

We introduce a process that allows one to test the validity of expert statements within the experimental data, that is, to achieve the querying of a relational database by a system expressed in the conceptual graph formalism. This process is based on the use of annotated conceptual graph patterns. When an expert statement appears not to be valid, a second-step objective is to refine it. This refinement consists of an automatic *exception rule learning* which provides unexpected knowledge in regard of previously established knowledge.

The examples given in this chapter have been designed using the CoGui tool (http://www.lirmm.fr/cogui/) and concern a concrete application in the domain of food quality.

BACKGROUND

Related Work

Handling exceptions is quite an old feature of artificial intelligence (Goodenough, 1975) that has been approached in various directions. In this project, we are concerned with the more specific theme of exception rules. Hussain (2000) explains very well the interest of exceptions as contradictions of common belief. Approaches for finding "interesting" rules are usually classified in two categories (Silberschatz, 1996): objective finding (as in Hussain, 2000), which relies on frequency based criteria and consists of identifying deviations among rules learnt from data, and subjective finding, which relies on belief based criteria and consists of identifying deviations to rules given by the user. Finding "unexpected" rules is part of the second category and can itself be subdivided in syntax based (Liu, 1997; Li, 2007 for a very recent work on sequence mining) and logic based (Padmanabhan, 1998; Padmanabhan, 2006) approaches.

Our approach is related to the latter, and more specifically to first-order rule learning techniques (Mitchell, 1997). However in the above approaches, rule learning is purely data driven and user knowledge is used as a filter, either in post-analysis (Liu, 1997; Sahar, 1999)

or in earlier stages (Padmanabhan, 1998, Wang, 2003), whereas we propose to find exception rules by trying variations – refinements – of the forms of the rules given by the experts, using an ontology that has been conceived with this specific purpose. Data are only used for rule verification. This reversed approach is relevant in domains characterized by a relatively high confidence in human expertise, and guarantees the learnt exceptions to be understandable and usable. This advantage is enforced by the graphical representation of the rules, expressed in the conceptual graph model.

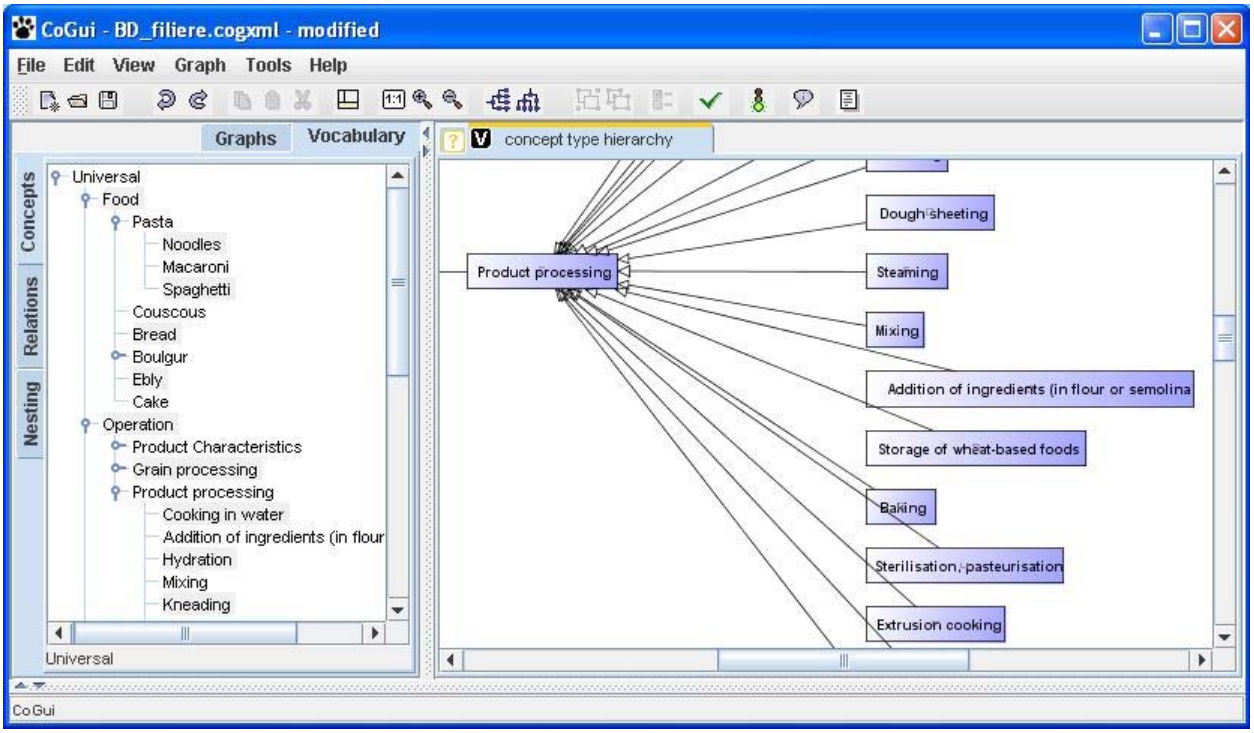
Conceptual Graph Model

The Conceptual Graph model (or CG) (Sowa, 1984), is a knowledge representation formalism based on labelled graphs. We use the formalization presented in (Mugnier, 2000). We will briefly present the support (terminological knowledge), the conceptual graphs (assertional knowledge), the specialization relation, and the rules.

The support provides the ground vocabulary used to build the knowledge base: types of concepts, instances of these types, and types of relations linking the concepts. The set of concept types is partially ordered by the "kind of" relation. Figure 1 presents a part of the set of concept types used in the application. An example of relation type is "undergoes" which is a binary relation allowing one to link a Food with an Operation (which are both concept types). The set of individual markers contains the instances of the concepts. For example, "months" can be an instance of Duration unit. The generic marker (denoted *) is a particular marker referring to an unspecified instance of a concept.

The conceptual graphs, built upon the support are composed of two kinds of vertices: (i) concept vertices (denoted in rectangles) which represent the entities, attributes, states, events; (ii) relation vertices (denoted in ovals) which express the nature of the relationship between concepts. The label of a concept vertex is a pair composed of a concept type and a marker (individual

Figure 1. A part of the set of concept types



or generic) of this type. The label of a relation vertex is its relation type.

Conceptual graph rules (Salvat, 1996) form an extension of the CG formalism, in which rules of the form "if A then B" are added to a knowledge base, where A and B are two simple conceptual graphs. The addition of rules considerably increases the expressivity of the language. Figure 3 shows a simple example of rule representing the information "If a food product undergoes cooking in water, then its vitamin content decreases".

The set of CGs is partially pre-ordered by the specialization relation (denoted ≤), which can be computed by the projection operation (a graph morphism allowing a restriction of the vertex labels): G2 ≤ G1 if and only if there is a projection of G1 into G2. The projection is a ground operation in the CG model since it allows the search for answers, which can be viewed as specializations of a query.

MAIN FOCUS

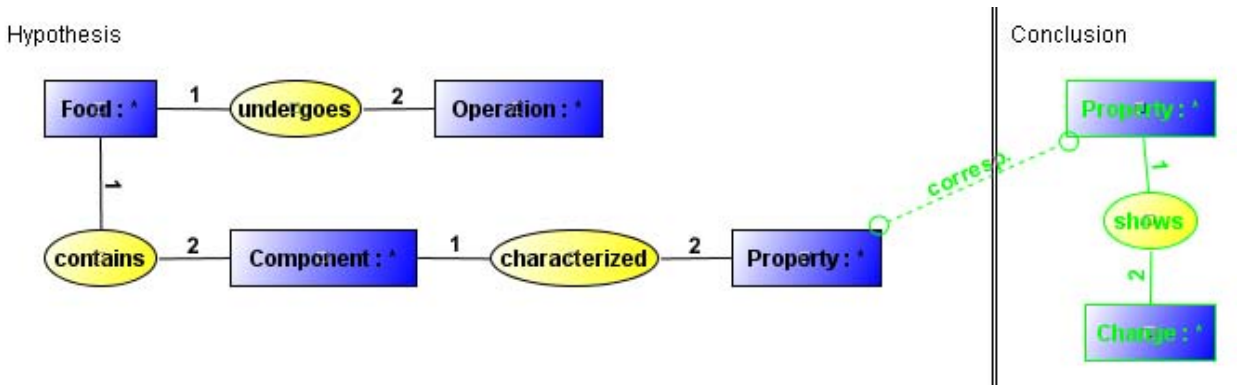
A preliminary step, presented in (Thomopoulos, 2007), was to automate the generation of an ontology, constituted of the set of concept types, by using the existing relational schema and data. This is processed in two main steps: the identification of high-level concept types, and the organization of these concept types into a hierarchy. Hence there are correspondancies between the vocabularies of both formalisms, which is assumed in the following.

Searching for Exceptions to Expert Rules within Experimental Data

The objective is to test whether the expert knowledge expressed as conceptual graph rules is valid within the experimental data of the relational database. This must be achieved without having to define manually, for each rule, the queries to be executed in the database to obtain this information. A confidence rate is computed for the tested rule and the data that constitute exceptions to the rule are identified and can be visualized by the user.

1. *Confidence* - Evaluating the validity of an expert rule within the data consists of calculating the proportion of data which satisfy both the hypothesis and the conclusion of the rule, among the data which satisfy the hypothesis of the rule. Let n_H be the number of data that satisfy the hypothesis and $n_{H\wedge C}$ the number of data that satisfy both the hypothesis and the conclusion. The confidence is $\tau = (n_{H\wedge C}) / n_H \times 100$, where n_H and $n_{H\wedge C}$ are the results of SQL queries counting the data. The problem to solve is the automation of the construction of these queries.
2. *Rule Patterns, Rule Instances and Associated Properties* - Although the expert rules can take various forms, they can be grouped into sets of

Figure 2. Example of rule pattern representing the information "If a food product, that contains a component characterized by a property, undergoes an operation, then this property changes"



rules which follow the same general form. The "general form" can itself be represented by a rule, called rule pattern. Its structure is identical to that of the expert rules that compose the set (called rule instances), but its concept vertices are more general. In other words, each instance rule has a hypothesis and a conclusion which are specializations of those of the rule pattern. For example, the rule instance of Figure 3 conforms to the rule pattern of Figure 2. Patterns help to constrain the search, and are a language bias.

The concept types that appear in a rule pattern provide a list of table names of the database. The hypothesis (respectively, the conclusion) of a rule pattern can be interpreted, within the database, as the formula of a query that selects the data satisfying the hypothesis (respectively, the conclusion). This formula simply specifies a query schema. It does not specify any particular selection criteria.

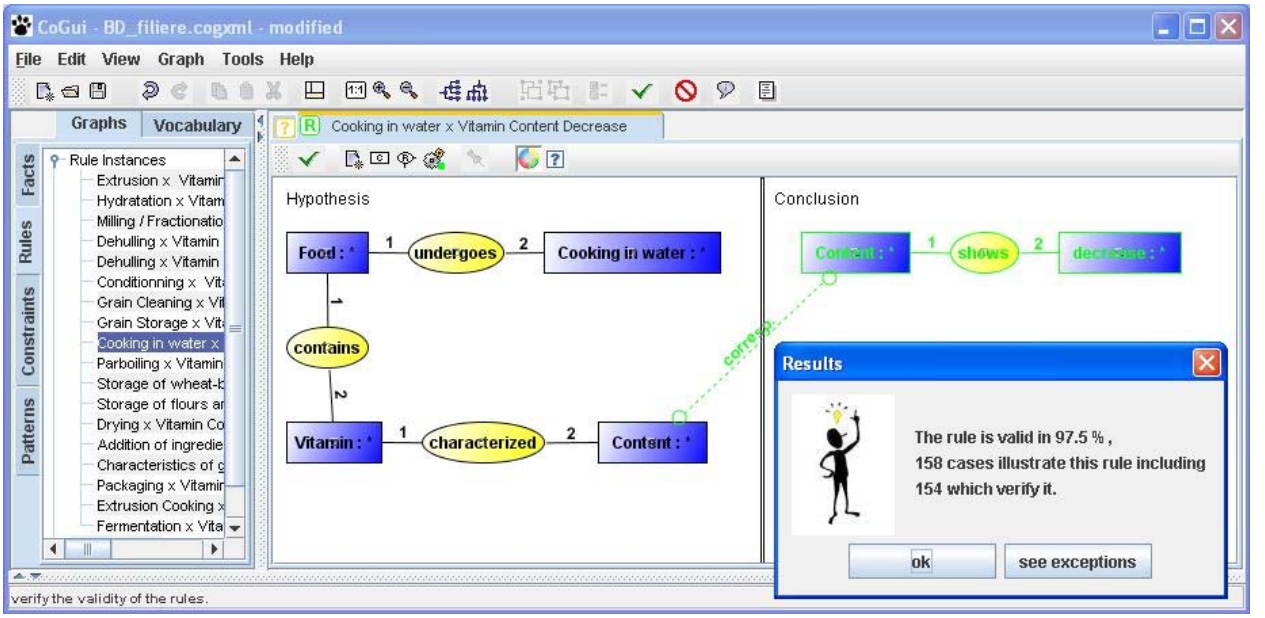
To allow the computation of the confidence and the link with the database, each rule pattern P is annotated with:

- a hypothesis query, that counts the tuples of the database satisfying the hypothesis of P;
- a hypothesis and conclusion query, that counts the tuples of the database satisfying both the hypothesis and the conclusion of P;
- for each of its concepts, the attribute of the database that contains the specializations of this concept expected in the rule instances conforming to P.

3. *Validation of a Rule Instance* - To test the validity of an expert rule, i.e. of a rule instance, two new queries are automatically built: a query that counts the data satisfying the hypothesis of the rule instance and a query that counts the data satisfying both the hypothesis and the conclusion.

These queries are composed of two parts: (i) the first part describes the schema of the query to be executed: this part corresponds to the query associated with the rule pattern, it is thus provided by the annotations of the rule pattern; (ii) the second part allows one to select exclusively the tuples which take the attribute values corresponding to the specializations that appear in the

Figure 3. Evaluation of the validity of an expert rule



rule instance. This part thus specifies selection criteria, which are automatically built by using, as selection attributes, the annotations of the rule pattern (database attributes) and as selection values, the labels of concepts in the rule instance to be evaluated.

The results of the queries are respectively n_H and $n_{H\wedge C}$, which allows the computation of the rule confidence. The rules whose confidence is strictly lower than 100% have exceptions within the database. For example the confidence τ of the rule of Figure 3 is equal to 97.5%. Confidence spread out from 73 to 100% in the application.

Unexpected Rule Mining

The aim of this part is to learn new unexpected rules corresponding to the exceptions identified in the previous stage. Three steps are proposed to achieve this objective, respectively negation, specialization and completion.

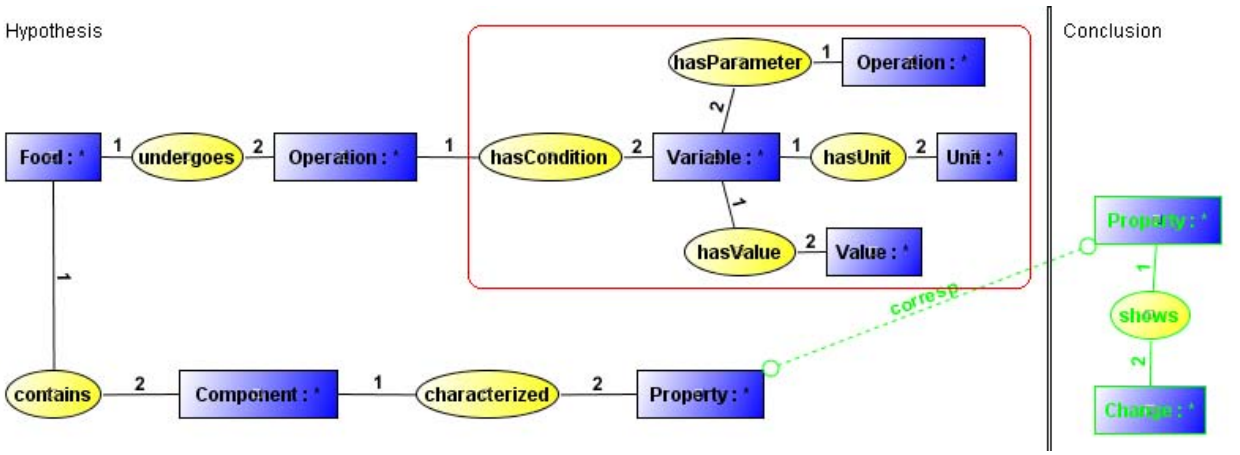
1. *Rule conclusion negation* – The exceptions to an expert rule do not satisfy the conclusion of this rule – this is the definition of such an exception. The first step of rule exception learning consists of negating the conclusion of the considered rule. The concepts to be negated are automatically identified through annotations of rule patterns. The images (through the projection operation) of rule pattern concepts that have the "negation" annotation are those to be negated in the rule instances. For instance, in the rule pattern of Figure 2, the concept of type Change has the "negation" annotation. After the negation step, the rule instance of Figure 3 is changed into: "if a food product undergoes cooking in water, then its vitamin content does **not** decrease" (it may increase or stagnate in the exceptions to the rule). The change in the rule conclusion is shown in Figure 5.
2. *Specialization* - Exceptions to generally admitted rules may be due to specific behaviors of particular sub-concepts. For instance, nutritional components are classified into heterogeneous families, whose specificities may explain the existence of exceptions. As in the previous step, the concepts to be specialized are identified through annotations of the rule patterns. The algorithmic principle of the specialization step is the following:
   - Let C be a concept, belonging to a rule R built in previous step, to be specialized (C is the image of a rule pattern concept having the "specialization" annotation).
   - Take the set (denoted S) of all the super-types more specific than the type of C in exceptions of R. In other words, we try to find a generalization of the concept types appearing in exceptions, that remains more specific than the concept type appearing in R.
   - Within these super-types, take the set of most general ones (this set is denoted E and is reduced to one element at the beginning, the type of C). For each type t of E, test if t or a sub-type of t appears in non-exceptions without appearing in exceptions. If it is the case, remove t from E and replace it by its most general sub-types in S, not comparable with types already present in E.
   - Go to next type t, until E remains stable. Create the exception rules by replacing the type of C by a type t of E, in the rule R.

   After the specialization step, the rule of Figure 3 has its "Vitamin" vertex specialized into "Liposoluble vitamin", as shown in Figure 5.
3. *Completion* - Exceptions to generally admitted rules may be due to the influence of other concepts, that are not taken into account in the rule. For instance, specific parameters of experimental conditions may explain the existence of exceptions. Predefined completion patterns (also annotated) are determined. Concepts of the rules to be precised by additional information are identified through annotations of the rule patterns. For example, the rule pattern of Figure 2 leads to a new rule pattern shown in Figure 4 (the completion pattern is encircled).

The algorithmic principle of the completion step is the following:

- For each rule built in previous step, fusion the concepts that are images of rule pattern concepts annotated by "completion" with corresponding concepts of completion patterns. The SQL query associated with the new rule will automatically be built by conjunction of the selection criteria

Figure 4. New rule pattern

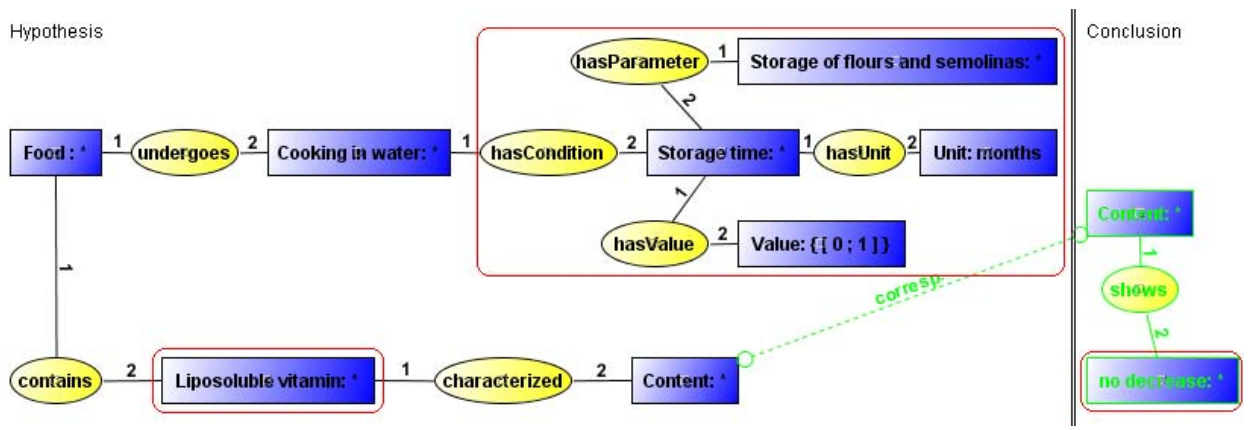


of the SQL queries associated with the merged graphs.

- Within the merged graph, a choice of more specialized concepts must be done (identified by annotations) to determine the most discriminating conditions that may explain the existence of exceptions. The following process is proposed. Let L be the list of experimental data that satisfy the hypothesis of the rule R which is being refined.

Within L, let L_e be the list of exceptions to R and L_v the list of data validating R. The concepts to be specialized will have their corresponding values in L examined. For each possible specialization of these concepts that appears in the values of L_e, an error is calculated, defined as the number of data in L_v that also take these values. Only the specializations that have the smaller error are retained.

Figure 5. The rule of Figure 3 after the negation, specialization and completion steps



After the three steps (negation, specialization and completion), the rule of Figure 3 produces the rule of Figure 5. It represents the information: "If a food product undergoes cooking in water, with a storage time of flours and semolina lower than one month, then its liposoluble vitamin content does not decrease".

FUTURE TRENDS

The long-term objective is to use the presented rules for decision-making: given a user's query that expresses a required goal, the issue is to determine which conditions allow one to achieve this goal, by identifying rules whose conclusions would satisfy the required goal, and whose hypotheses would provide sufficient conditions to obtain it.

This objective requires a deep knowledge of the domain rules. The approach presented in this chapter for exception rule learning contributes to this aim.

CONCLUSION

Given two heterogeneous kinds of information available on a domain (generic expert rules, and detailed experimental results) represented in two distinct formalisms (respectively the conceptual graph model and the relational model), in this article we proposed two stages for the construction of a deep expertise on the domain: (i) the evaluation of the validity of expert knowledge within the experimental data. This stage is based on the notion of rule pattern in the conceptual graph formalism, associated with a corresponding SQL query schema in the relational formalism. The evaluation of a rule instance that conforms to a given rule pattern is then processed by completing the query schema associated with the pattern by selection criteria specific to the considered rule instance; (ii) learning of unexpected and more precise rules that correspond to exceptions to rules, detected in the previous stage. These stages are automatic, which is allowed by annotations of the rule patterns.

The proposed methodology thus relies on the cooperation of the two kinds of information and the two heterogeneous formalisms.

REFERENCES

Bos, C., Botella, B., & Vanheeghe, P. (1997). Modeling and Simulating Human Behaviors with Conceptual Graphs. In D. Lukose (Ed.), *LNAI: Vol. 1257. Conceptual Structures: Fulfilling Peirce's Dream* (pp. 275-289). Heidelberg, Germany: Springer.

Goodenough, J.B. (1975). Exception handling: issues and a proposed notation. *Communications of the ACM* 12(18), 683-693.

Hussain, F., Liu, H., Suzuki, E., & Lu, H. (2000). Exception Rule Mining with a Relative Interestingness Measure. In T. Terano (Ed.), *LNCS: Vol. 1805. Knowledge Discovery and Data Mining: Current Issues and New Applications* (pp. 86-97). Heidelberg, Germany: Springer.

Li, D.H., Laurent, A., & Poncelet, P. (2007, July). *Towards Unexpected Sequential Patterns*. Paper presented at the workshop on inductive databases, Grenoble, France. Retrieved July 14, 2007, from http://afia2007.imag.fr/programme/bdi07ext_li.pdf

Liu, B., Hsu, W., & Chen, S. (1997). Using general impressions to analyze discovered classification rules. In D. Heckerman (Ed.), *Third International Conference on Knowledge Discovery and Data mining* (pp. 31-36). Menlo Park, CA: AAAI Press.

Mitchell, T. (1997). Learning Sets of Rules. In *Machine Learning* (pp. 284-316). Blacklick, OH: McGraw Hill.

Mugnier, M.L. (2000). Knowledge Representation and Reasoning based on Graph Homomorphism. In B. Ganter (Ed.), *LNAI: Vol 1867. Conceptual Structures: Logical, Linguistic, and Computational Issues* (pp. 172-192). Heidelberg, Germany: Springer.

Padmanabhan, B., & Tuzhilin, A. (1998). A belief-driven method for discovering unexpected patterns. In R. Agrawal (Ed.), *Fourth International Conference on Knowledge Discovery and Data mining* (pp. 94-100). Menlo Park, CA: AAAI Press.

Padmanabhan, B., & Tuzhilin, A. (2006). On characterization and discovery of minimal unexpected patterns in rule discovery. *IEEE Transactions on Knowledge and Data Engineering* 18(2), 202-216.

Sahar, S. (1999). Interestingness via what is not interesting. In S. Chaudhuri (Ed.), *Fifth International Conference on Knowledge Discovery and Data mining* (pp. 332-336). New York, NY: ACM Press.

Salvat, E., & Mugnier, M.L. (1996). Sound and Complete Forward and Backward Chaining of Graph Rules. In P.W. Eklund (Ed.), *LNAI: Vol. 1115. Conceptual Structures: Knowledge Representations as Interlingua* (pp.248-262). Heidelberg, Germany: Springer.

Silberschatz, A., & Tuzhilin, A. (1996). What makes patterns interesting in knowledge discovery systems. *IEEE Transactions on Knowledge and Data Engineering* 8(6), 970-974.

Sowa, J.F. (1984). *Conceptual Structures: Information Processing in Mind and Machine*. Reading, MA: Addison-Wesley.

Thomopoulos, R., Baget, J.F., & Haemmerlé, O. (2007). Conceptual Graphs as Cooperative Formalism to Build and Validate a Domain Expertise. In S. Polovina (Ed.), *LNAI: Vol. 4604. Conceptual Structures: Knowledge Architectures for Smart Applications*. Heidelberg, Germany: Springer.

Wang, K., Jiang, Y., & Lakshmanan (2003). Mining Unexpected Rules by Pushing User Dynamics. In L. Getoor (Ed.), *Ninth International Conference on Knowledge Discovery and Data mining* (pp. 246-255). New York, NY: ACM Press.

KEY TERMS

CG Rule Instance: A conceptual graph rule that is obtained by specialization of a CG rule pattern.

CG Rule Pattern: A conceptual graph rule that contains generic vertices and translates the general form common to several pieces of information.

Conceptual Graphs: A knowledge representation and reasoning model from artificial intelligence, of the family of semantic networks.

Data and Expert Knowledge Cooperation: Mixed information handling approach, both data driven and symbolic-rooted through expert statements.

Knowledge Refinement: Definition of more specific knowledge than the initially available one.

Rule Validation: Checking the absence of exceptions to a given rule within raw data.

Unexpected Knowledge: Knowledge that constitutes an exception to commonly admitted information on a domain.

Learning from Data Streams

João Gama
University of Porto, Portugal

Pedro Pereira Rodrigues
University of Porto, Portugal

INTRODUCTION

In the last two decades, machine learning research and practice has focused on batch learning usually with small datasets. In batch learning, the whole training data is available to the algorithm that outputs a decision model after processing the data eventually (or most of the times) multiple times. The rationale behind this practice is that examples are generated at random accordingly to some stationary probability distribution. Also, most learners use a greedy, hill-climbing search in the space of models.

What distinguishes current data sets from earlier ones are the continuous flow of data and the automatic data feeds. We do not just have people who are entering information into a computer. Instead, we have computers entering data into each other. Nowadays there are applications in which the data is modelled best not as persistent tables but rather as transient data streams. In some applications it is not feasible to load the arriving data into a traditional DataBase Management Systems (DBMS), and traditional DBMS are not designed to directly support the continuous queries required in these application (Babcock et al., 2002). These sources of data are called *Data Streams*. There is a fundamental difference between learning from small datasets and large datasets. As pointed-out by some researchers (Brain & Webb, 2002), current learning algorithms emphasize variance reduction. However, learning from large datasets may be more effective when using algorithms that place greater emphasis on bias management.

Algorithms that process data streams deliver approximate solutions, providing a fast answer using few memory resources. They relax the requirement of an exact answer to an approximate answer within a small error range with high probability. In general, as the range of the error decreases the space of computational resources goes up. In some applications, mostly database oriented, an approximate answer should be within an admissible error margin. Some results on tail inequalities provided by statistics are useful to accomplish this goal.

LEARNING ISSUES: ONLINE, ANYTIME AND REAL-TIME LEARNING

The challenge problem for data mining is the ability to permanently maintain an accurate decision model. This issue requires learning algorithms that can modify the current model whenever new data is available at the rate of data arrival. Moreover, they should forget older information when data is out-dated. In this context, the assumption that examples are generated at random according to a stationary probability distribution does not hold, at least in complex systems and for large time periods. In the presence of a non-stationary distribution, the learning system must incorporate some form of forgetting past and outdated information. Learning from data streams require incremental learning algorithms that take into account concept drift. Solutions to these problems require new sampling and randomization techniques, and new approximate, incremental and decremental algorithms. In (Hulten & Domingos, 2001), the authors identify desirable properties of learning systems that are able to mine continuous, high-volume, open-ended data streams as they arrive: *i*) incrementality, *ii*) online learning, *iii*) constant time to process each example using fixed memory, *iv*) single scan over the training data, and *v*) tacking drift into account.

Examples of learning algorithms designed to process open-ended streams include predictive learning: Deci-

sion Trees (Domingos & Hulten, 2000; Hulten et al., 2001; Gama et al., 2005, 2006), Decision Rules (Ferrer et al., 2005); descriptive learning: variants of k-Means Clustering (Zhang et al., 1996; Sheikholeslami et al., 1998), Clustering (Guha et al., 1998; Aggarwal et al., 2003), Hierarchical Time-Series Clustering (Rodrigues et al., 2006); Association Learning: Frequent Itemsets Mining (Jiang & Gruemwald, 2006), Frequent Pattern Mining (Jin & Agrawal 2007); Novelty Detection (Markou & Singh, 2003; Spinosa et al. 2007); Feature Selection (Sousa et al., 2006), etc.

All these algorithms share some common properties. They process examples at the rate they arrive using a single scan of data and fixed memory. They maintain a decision model at any time, and are able to adapt the model to the most recent data.

Incremental and Decremental Issues

The ability to update the decision model whenever new information is available is an important property, but it is not enough. Another required operator is the ability to *forget* past information (Kifer et al., 2004). Some data stream models allow delete and update operators. For example, *sliding windows* models require the forgetting of old information. In these situations the incremental property is not enough. Learning algorithms need forgetting operators that reverse learning: decremental unlearning (Cauwenberghs & Poggio, 2000).

Cost-Performance Management

Learning from data streams require to update the decision model whenever new information is available. This ability can improve the *flexibility* and *plasticity* of the algorithm in fitting data, but at some *cost* usually measured in terms of resources (time and memory) needed to update the model. It is not easy to define where is the trade-off between the benefits in flexibility and the cost for model adaptation. Learning algorithms exhibit different profiles. Algorithms with strong variance management are quite efficient for small training sets. Very simple models, using few free-parameters, can be quite efficient in variance management, and effective in incremental and decremental operations (for example naive Bayes (Domingos & Pazzani, 1997)) being a natural choice in the sliding windows framework. The main problem with simple approaches is the boundary in generalization performance they can achieve; they are limited by high bias. Complex tasks requiring more complex models increase the search space and the cost for structural updating. These models require efficient control strategies for the trade-off between the gain in performance and the cost of updating.

Monitoring Learning

Whenever data flows over time, it is highly improvable the assumption that the examples are generated at random according to a stationary probability distribution (Basseville & Nikiforov, 1993). At least in complex systems and for large time periods, we should expect changes (smooth or abrupt) in the distribution of the examples. A natural approach for these *incremental tasks* is *adaptive learning algorithms*, incremental learning algorithms that take into account concept drift.

Change Detection

Concept drift (Klinkenberg, 2004, Aggarwal, 2007) means that the concept about which data is being collected may shift from time to time, each time after some minimum permanence. Changes occur over time. The evidence for changes in a concept is reflected in some way in the training examples. Old observations, that reflect the behaviour of nature in the past, become irrelevant to the current state of the phenomena under observation and the learning agent must forget that information. The nature of change is diverse. Changes may occur in the context of learning, due to changes in hidden variables, or in the characteristic properties of the observed variables.

Most learning algorithms use blind methods that adapt the decision model at regular intervals without considering whether changes have really occurred. Much more interesting are explicit change detection mechanisms. The advantage is that they can provide meaningful description (indicating change-points or small time-windows where the change occurs) and quantification of the changes. They may follow two different approaches:

- Monitoring the evolution of performance indicators adapting techniques used in Statistical Process Control (Gama et al., 2004).
- Monitoring distributions on two different time windows (Kiffer et al., 2004). The method monitors the evolution of a distance function between

two distributions: data in a *reference window* and in a current window of the most recent data points.

The main research issue is to develop methods with fast and accurate detection rates with few false alarms. A related problem is: how to incorporate change detection mechanisms inside learning algorithms. Also, the level of *granularity* of decision models is a relevant property (Gaber et al., 2004), because it can allow partial, fast and efficient updating in the decision model instead of rebuilding a complete new model whenever a change is detected. Finally, the ability to recognize seasonal and re-occurring patterns is an open issue.

FUTURE TRENDS IN LEARNING FROM DATA STREAMS

Streaming data offers a symbiosis between Streaming Data Management Systems and Machine Learning. The techniques developed to estimate synopsis and sketches require counts over very high dimensions both in the number of examples and in the domain of the variables. On one hand, the techniques developed in data streams management systems can provide tools for designing Machine Learning algorithms in these domains. On the other hand, Machine Learning provides compact descriptions of the data than can be useful for answering queries in DSMS. What are the current trends and directions for future research in learning from data streams?

Issues on incremental learning and forgetting are basic issues in stream mining. In most applications, we are interested in maintaining a decision model consistent with the current status of the nature. This has led us to the sliding window models where data is continuously inserted and deleted from a window. So, learning algorithms must have operators for incremental learning and forgetting. Closed related are change detection issues. Concept drift in the predictive classification setting is a well studied topic (Klinkenberg, 2004). In other scenarios, like clustering, very few works address the problem. The main research issue is how to incorporate change detection mechanisms in the learning algorithm for different paradigms. Another relevant aspect of any learning algorithm is the hypothesis evaluation criteria and metrics. Most of evaluation methods and metrics were designed for the static case and provide a single measurement about the quality of the hypothesis. In the streaming context, we are much more interested in how the evaluation metric evolves over time. Results from the *sequential statistics* (Wald, 1947) may be much more appropriate.

CONCLUSION

Learning from data streams is an increasing research area with challenging applications and contributions from fields like data bases, learning theory, machine learning, and data mining. Sensor networks, scientific data, monitoring processes, web analysis, traffic logs, are examples of real-world applications were stream algorithms have been successful applied. Continuously learning, forgetting, self-adaptation, and self-reaction are main characteristics of any intelligent system. They are characteristic properties of stream learning algorithms.

REFERENCES

Aggarwal, C., Han, J., Wang, J., & Yu, P. (2003). A framework for clustering evolving data streams. In *VLDB 2003, Proceedings of Twenty-Ninth International Conference on Very Large Data Bases* (pp. 81-92). Morgan Kaufmann.

Aggarwal, C. C. (2007). A Survey of Change Diagnosis, In C. Aggarwal (Ed.), *Data Streams: Models and Algorithms* (pp. 85-102). Springer.

Babcock, B., Babu, S., Datar, M., Motwani., R., & Widom. J. (2002) Models and issues in data stream systems. In Lucian Popa (Ed.), *Proceedings of the 21st Symposium on Principles of Database Systems* (pp. 1-16). ACM Press.

Basseville M., & Nikiforov I. (1993). *Detection of Abrupt Changes - Theory and Application*. Prentice-Hall.

Brain D., & Webb G. (2002). The need for low bias algorithms in classification learning from large data sets. In T.Elomaa, H.Mannila, and H.Toivonen (Eds.), *Principles of Data Mining and Knowledge Discovery PKDD-02* (pp. 62-73). LNAI 2431, Springer Verlag.

Cauwenberghs G., & Poggio T. (2000). Incremental and decremental support vector machine learning. In T. K. Leen, T. G. Dietterich and V. Tresp (Eds.), *Proceedings of the 13th Neural Information Processing Systems* (pp. 409-415). MIT Press.

Domingos P., & Hulten G. (2000). Mining High-Speed Data Streams. In *Proceedings of the ACM Sixth International Conference on Knowledge Discovery and Data Mining* (pp. 71-80). ACM Press.

Domingos, P., & Pazzani, M. (1997). On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, *29*, 103-130.

Ferrer, F., & Aguilar, J., & Riquelme, J. (2005). Incremental rule learning and border examples selection from numerical data streams. *Journal of Universal Computer Science*, *11* (8), 1426-1439.

Gaber, M., & Zaslavsky, A., & Krishnaswamy, S. (2004). Resource-Aware Knowledge Discovery in Data Streams. In *International Workshop on Knowledge Discovery in Data Streams; ECML-PKDD04* (pp. 32-44). Tech. Report, University of Pisa.

Gama, J., Fernandes, R., & Rocha, R. (2006). Decision trees for mining data streams. *Intelligent Data Analysis*, *10* (1), 23-45.

Gama, J., Medas, P., Castillo, G., & Rodrigues, P. (2004). Learning with drift detection. In A. L. C. Bazzan and S. Labidi (Eds.), *Proceedings of the 17th Brazilian Symposium on Artificial Intelligence* (pp. 286-295). LNAI 3171. Springer.

Gama, J., Medas, P., & Rodrigues, P. (2005). Learning decision trees from dynamic data streams. In H. Haddad, L. Liebrock, A. Omicini, and R. Wainwright (Eds.), *Proceedings of the 2005 ACM Symposium on Applied Computing* (pp. 573-577). ACM Press.

Guha, S., Rastogi, R., & Shim, K. (1998). CURE: An efficient clustering algorithm for large databases. In L. Haas and A. Tiwary (Eds.), *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data* (pp. 73-84). ACM Press.

Hulten, G., & Domingos, P. (2001). Catching up with the data: research issues in mining data streams. In *Proceedings of Workshop on Research issues in Data Mining and Knowledge Discovery*.

Hulten, G., Spencer, L., & Domingos, P. (2001). Mining time-changing data streams. In *Proceedings of the 7th ACM SIGKDD International conference on Knowledge discovery and data mining* (pp. 97-106). ACM Press.

Jiang, N., & Gruemwald, L. (2006). Research Issues in Data Stream Association Rule Mining, *SIGMOD Record*, *35*, 14-19.

Jin, R., & Agrawal, G. (2007). Frequent Pattern Mining in Data Sreams. In C. Aggarwal (Ed.), *Data Streams: Models and Algorithms* (pp. 61-84). Springer.

Kargupta, H., Joshi, A., Sivakumar, K., & Yesha, Y. (2004). *Data Mining: Next Generation Challenges and Future Directions*. AAAI Press and MIT Press.

Kiffer, D., Ben-David, S., & Gehrke, J. (2004). Detecting change in data streams. In *VLDB 04: Proceedings of the 30th International Conference on Very Large Data Bases* (pp. 180-191). Morgan Kaufmann Publishers Inc.

Klinkenberg, R. (2004). Learning drifting concepts: Example selection vs. example weighting. *Intelligent Data Analysis*, *8* (3), 281–300.

Markou, M., & Singh, S. (2003). Novelty Detection: a Review. *Signal Processing*, *83* (12), 2499–2521.

Motwani, R., & Raghavan, P. (1997). *Randomized Algorithms*. Cambridge University Press.

Muthukrishnan, S. (2005). *Data streams: algorithms and applications*. Now Publishers.

Rodrigues, P. P., Gama, J., & Pedroso, J. P. (2006). ODAC: Hierarchical Clustering of Time Series Data Streams. In J. Ghosh, D. Lambert, D. Skillicorn, and J. Srivastava (Eds.), *Proceedings of the Sixth SIAM International Conference on Data Mining* (pp. 499-503). SIAM.

Sheikholeslami, G., Chatterjee, S., & Zhang, A. (1998). WaveCluster: A multi-resolution clustering approach for very large spatial databases. In *Proceedings of the Twenty-Fourth International Conference on Very Large Data Bases* (pp. 428-439). ACM Press.

Sousa, E., Traina, A., Traina Jr, C. & Faloutsos, C. (2006). Evaluating the Intrinsic Dimension of Evolving Data Streams, In *Proceedings of the 2006 ACM Symposium on Applied Computing* (pp. 643-648). ACM Press.

Spinosa, E., Carvalho, A., & Gama, J., (2007). OLLINDA: A cluster-based approach for detecting novelty and concept drift in data streams. In *Proceedings of the 2007 ACM Symposium on Applied Computing* (pp. 448-452). ACM Press.

Wald, A. (1947). *Sequential Analysis*. John Wiley and Sons, Inc.

Zhang, T., Ramakrishnan, R., & Livny, M. (1996). BIRCH: An efficient data clustering method for very large databases. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data* (pp. 103-114). ACM Press.

KEY TERMS

Adaptive Learning: Learning algorithms that self-modify their model by incorporating new information and/or forgetting outdated information.

Association rules: Rules that describe events that are frequently observed together.

Concept Drift: Any change in the distribution of the examples that describe a concept.

Data Mining: The process of extraction of useful information in large Data Bases.

Data Stream: Continuous flow of data eventually at high speed

Decremental Learning: Learning that modifies that current decision model by forgetting the oldest examples.

Feature selection: Process focussing on selecting the attributes of a dataset, which are relevant for the learning task.

Incremental Learning: Learning that modifies that current decision model using only the most recent examples

Machine Learning: Programming computers to optimize a performance criterion using example data or past experience.

Novelty Detection: The process of identification new and unknown patterns that a machine learning system is not aware of during training.

Online Learning: Learning by a sequence of predictions followed by rewards from the environment.

Learning Kernels for Semi-Supervised Clustering

Bojun Yan
George Mason University, USA

Carlotta Domeniconi
George Mason University, USA

INTRODUCTION

As a recent emerging technique, semi-supervised clustering has attracted significant research interest. Compared to traditional clustering algorithms, which only use unlabeled data, semi-supervised clustering employs both unlabeled and supervised data to obtain a partitioning that conforms more closely to the user's preferences. Several recent papers have discussed this problem (Cohn, Caruana, & McCallum, 2003; Bar-Hillel, Hertz, Shental, & Weinshall, 2003; Xing, Ng, Jordan, & Russell, 2003; Basu, Bilenko, & Mooney, 2004; Kulis, Dhillon, & Mooney, 2005).

In semi-supervised clustering, limited supervision is provided as input. The supervision can have the form of labeled data or pairwise constraints. In many applications it is natural to assume that pairwise constraints are available (Bar-Hillel, Hertz, Shental, & Weinshall, 2003; Wagstaff, Cardie, Rogers, & Schroedl, 2001). For example, in protein interaction and gene expression data (Segal, Wang, & Koller, 2003), pairwise constraints can be derived from the background domain knowledge. Similarly, in information and image retrieval, it is easy for the user to provide feedback concerning a qualitative measure of similarity or dissimilarity between pairs of objects. Thus, in these cases, although class labels may be unknown, a user can still specify whether pairs of points belong to the same cluster (Must-Link) or to different ones (Cannot-Link). Furthermore, a set of classified points implies an equivalent set of pairwise constraints, but not vice versa. Recently, a kernel method for semi-supervised clustering has been introduced (Kulis, Dhillon, & Mooney, 2005). This technique extends semi-supervised clustering to a kernel space, thus enabling the discovery of clusters with non-linear boundaries in input space. While a powerful technique, the applicability of a kernel-based semi-supervised clustering approach is limited in practice, due to the critical settings of kernel's parameters. In fact, the chosen parameter values can largely affect the quality of the results. While solutions have been proposed in supervised learning to estimate the optimal kernel's parameters, the problem presents open challenges when no labeled data are provided, and all we have available is a set of pairwise constraints.

BACKGROUND

In the context of supervised learning, the work in (Chapelle & Vapnik) considers the problem of automatically tuning multiple parameters for a support vector machine. This is achieved by minimizing the estimated generalization error achieved by means of a gradient descent approach over the set of parameters. In (Wang, Xu, Lu, & Zhang, 2002), a Fisher discriminant rule is used to estimate the optimal spread parameter of a Gaussian kernel. The authors in (Huang, Yuen, Chen & Lai, 2004) propose a new criterion to address the selection of kernel's parameters within a kernel Fisher discriminant analysis framework for face recognition. A new formulation is derived to optimize the parameters of a Gaussian kernel based on a gradient descent algorithm. This research makes use of labeled data to address classification problems. In contrast, the approach we discuss in this chapter optimizes kernel's parameters based on unlabeled data and pairwise constraints, and aims at solving clustering problems. In the context of semi-supervised clustering, (Cohn, Caruana, & McCallum, 2003) uses gradient descent combined with a weighted Jensen-Shannon divergence for EM clustering. (Bar-Hillel, Hertz, Shental, & Weinshall, 2003) proposes a Redundant Component Analysis (RCA) algorithm that uses only must-link constraints to learn a Mahalanobis distance. (Xing, Ng, Jordan, & Russell, 2003) utilizes both must-link and cannot-link constraints

to formulate a convex optimization problem which is local-minima-free. (Segal, Wang, & Koller, 2003) proposes a unified Markov network with constraints. (Basu, Bilenko, & Mooney, 2004) introduces a more general HMRF framework, that works with different clustering distortion measures, including Bregman divergences and directional similarity measures. All these techniques use the given constraints and an underlying (linear) distance metric for clustering points in input space. (Kulis, Dhillon, & Mooney, 2005) extends the semi-supervised clustering framework to a non-linear kernel space. However, the setting of the kernel's parameter is left to manual tuning, and the chosen value can largely affect the results. The selection of kernel's parameters is a critical and open problem, which has been the driving force behind our research work.

MAIN FOCUS

In kernel-based learning algorithms, a kernel function $K(x_i, x_j)$ allows the calculation of dot products in feature space without knowing explicitly the mapping function. It is important that the kernel function in use conforms to the learning target. For classification, the distribution of data in feature space should be correlated to the label distribution. Similarly, in semi-supervised clustering, one wishes to learn a kernel that maps pairs of points subject to a must-link constraint close to each other in feature space, and maps points subject to a cannot-link constraint far apart in feature space. The authors in (Cristianini, Shawe-Taylor, & Elisseeff) introduce the concept of kernel alignment to measure the correlation between the groups of data in feature space and the labeling to be learned. In (Wang, Xu, Lu & Zhang), a Fisher discriminant rule is used to estimate the optimal spread parameter of a Gaussian kernel. The selection of kernel's parameters is indeed a critical problem. For example, empirical results in the literature have shown that the value of the spread parameter σ of a Gaussian kernel can strongly affect the generalization performance of an SVM. Values of σ which are too small or too large lead to poor generalization capabilities. When $\sigma \to 0$, the kernel matrix becomes the identity matrix. In this case, the resulting optimization problem gives Lagrangians which are all 1s, and therefore every point becomes a support vector. On the other hand, when $\sigma \to \infty$, the kernel matrix has entries all equal to 1, and thus each point in feature space is maximally similar to each other. In both cases, the machine will generalize very poorly. The problem of setting kernel's parameters, and of finding in general a proper mapping in feature space, is even more difficult when no labeled data are provided, and all we have available is a set of pairwise constraints. In our research we utilize the given constraints to derive an optimization criterion to automatically estimate the optimal kernel's parameters. Our approach integrates the constraints into the clustering objective function, and optimizes the kernel's parameters iteratively while discovering the clustering structure. Specifically, we steer the search for optimal parameter values by measuring the amount of must-link and cannot-link constraint violations in feature space. Following the method proposed in (Basu, Bilenko, & Mooney, 2004; Bilenko, Basu, & Mooney), we scale the penalty terms by the distances of points, that violate the constraints, in feature space. That is, for violation of a must-link constraint (x_i, x_j), the larger the distance between the two points x_i and x_j in feature space, the larger the penalty; for violation of a cannot-link constraint (x_i, x_j), the smaller the distance between the two points x_i and x_j in feature space, the larger the penalty. Considering the Gaussian kernel function

$$K(x_i, x_j) = \exp\left(-\left\|x_i - x_j\right\|/(2\sigma^2)\right)$$

and adding the constraint

$$\sum_{x_i \in X} \left\|\phi(x_i) - \phi(x_r)\right\|^2 \geq Const$$

(where x_r is a point randomly selected from data set X) to avoid the trivial minimization of the objective function $J_{kernel-obj}$, we obtain the following function:

$$J_{kernel-obj} = \sum_{c=1}^{k} \sum_{x_i, x_j \in \pi_c} \frac{1 - K(x_i, x_j)}{|\pi_c|} + \sum_{(x_i, x_j) \in ML, l_i \neq l_j} 2w_{ij}(1 - K(x_i, x_j)) + \sum_{(x_i, x_j) \in CL, l_i = l_j} 2\overline{w}_{ij}(K(x_i, x_j) - K(x', x'')) - \left(\sum_{x_i \in X} 2(1 - K(x_i, x_r)) - Const\right)$$

where LM is the set of must-link constraints, LC is the set of cannot-link constraints, π_c represents the C^{th} cluster, x' and x'' are the farthest points in feature space, w_{ij} and $\overline{w}_{ij}$ are the penalty costs for violating a must-link and a cannot-link constraints respectively, and l_i represents the cluster label of x_i. The resulting minimization problem is non-convex.

EM-Based Strategy

To minimize the objective function $J_{kernel-obj}$, we use an EM-based strategy. The clusters are initialized through the mechanism proposed in (Kulis, Dhillon, & Mooney, 2005): we take the transitive closure of the constraints to form neighborhoods, and then perform a farthest-first traversal on these neighborhoods to get the K initial clusters.

E-step: The algorithm assigns data points to clusters so that the objective function $J_{kernel-obj}$ is minimized. Since the objective function integrates the given must-link and cannot-link constraints, the optimal assignment of each point is the one that minimizes the distance between the point and its cluster representative in feature space (first term of $J_{kernel-obj}$), while incurring a minimal penalty for constraint violations caused by this assignment (second and third terms of $J_{kernel-obj}$). The fourth term of $J_{kernel-obj}$ is constant during the assignment of data points in each iteration. When updating the cluster assignment of a given point, the assignment for the other points is kept fixed (Besag, 1986; Zhang, Brady, & Smith, 2001). During each iteration, data points are re-ordered randomly. The process is repeated until no change in point assignment occurs.

M-step: The algorithm re-computes the cluster representatives. Constraints are not used in this step. Therefore, only the first term of $J_{kernel-obj}$ is minimized. We note that all the steps so far are executed with respect to the current feature space. We now optimize the feature space by optimizing the kernel parameter σ. To this extent, the gradient descent rule is applied to update the parameter σ of the Gaussian kernel:

$$\sigma^{(new)} = \sigma^{(old)} - \rho \frac{\partial J_{Kernel-obj}}{\partial \sigma},$$

where ρ is a scalar step length parameter optimized via a line-search method.

Given the non-convex formulation, EM-based optimization can only reach a local optimal solution. Thus, the optimum parameter obtained by solving the adaptive kernel formulation may not be global. Nevertheless, obtaining global optimal parameter for clustering is very hard in general.

FUTURE TRENDS

Semi-supervised clustering is becoming an increasingly important research area in the fields of machine learning and data mining. Targeting at utilizing the limited pairwise constraints to improve clustering results, researchers are exploring different ways to maximize the utility of the given side information. One promising direction is the active learning of constraints which are most informative. Learning kernel functions (e.g., polynomial), as well as combinations of different types of kernels is also an interesting route to explore. Finally, we need to develop general methods to integrate constraints into the current existing clustering algorithms.

CONCLUSION

The proposed adaptive semi-supervised Kernel-KMeans algorithm can integrate the given constraints with the kernel function, and is able to automatically embed, during the clustering process, the optimal non-linear similarity within the feature space. As a result, the proposed algorithm is capable of discovering clusters with non-linear boundaries in input space with high accuracy. This technique enables the practical utilization of powerful kernel-based semi-supervised clustering approaches by providing a mechanism to automatically set the involved critical parameters.

REFERENCES

Bar-Hillel, A., Hertz, T., Shental, N., & Weinshall, D. (2003). Learning distance functions using equivalence relations. *International Conference on Machine Learning*, 11-18.

Basu, S., Bilenko, M., & Mooney, R.J. (2004). A probabilistic framework for semi-supervised clustering. *International Conference on Knowledge Discovery and Data Mining*.

Besag, J. (1986). On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society, Series B (Methodological)*.

Bilenko, M., Basu, S., & Mooney, R.J. (2004). Integrating constraints and Metric Learning in semi-supervised

clustering. *International Conference on Machine Learning*.

Chapelle, O., & Vapnik, V. (2002). Choosing Mutiple Parameters for Support Vector Machines. *Machine Learning, 46*(1), 131-159.

Cohn, D., Caruana, R., & McCallum, A. (2003). *Semi-supervised clustering with user feedback*. TR2003-1892, Cornell University.

Huang, J., Yuen, P.C., Chen, W.S., & Lai, J. H. (2004). *Kernel Subspace LDA with optimized Kernel Parameters on Face Rcognition*. The sixth IEEE International Conference on Automatic Face and Gesture Recognition.

Kulis, B., Basu, S., Dhillon, I., & Moony, R. (2005). Semi-supervised graph clustering: A kernel approach. *International Conference on Machine Learning*.

Segal, E., Wang, H., & Koller, D. (2003). Discovering molecular pathways from protein interaction and gene expression data. *Bioinformatics*.

Xing, E.P., Ng, A.Y., Jordan, M.I., & Russell, S. (2003). Distance metric learning, with application to clustering with side-information. *Advances in Neural Information Processing Systems 15*.

Wang, W., Xu, Z., Lu W., & Zhang, X. (2002). Determination of the spread parameter in the Gaussian Kernel for classification and regression. *Neurocomputing, 55*(3), 645.

Wagstaff, K., Cardie, C., Rogers, S., & Schroedl, S. (2001). Constrained K-Means clustering with background knowledge. *International Conference on Machine Learning*, 577-584.

Zhang, Y., Brady, M., & Smith, S. (2001). Hidden Markov random field model and segmentation of brain MR images. *IEEE Transactions on Medical Imaging*. 2001.

KEY TERMS

Adaptive Kernel Method: A new optimization criterion that can automatically determine the optimal parameter of an RBF kernel, directly from the data and the given constraints.

Cannot-Link Constraints: Two points that must be assigned to different clusters.

Clustering: The process of grouping objects into subsets, such that those within each cluster are more closely related to one another than objects assigned to different clusters, according to a given similarity measure.

EM: Expectation Maximization. The algorithm iteratively calculates the expectation of the likelihood function (E-Step) and maximizes the likelihood function (M-Step) to find the maximum likelihood estimates of parameters.

Kernel Kmeans: The clustering algorithm that first maps the data to a feature space, and then use the kernel trick to compute Euclidean distances in feature space to assign data points to the cluster with the nearest centroid.

Kernel Methods: Pattern analysis techniques that work by embedding the data into a high dimensional vector space, and by detecting linear relations in that space. A kernel function takes care of the embedding.

KMeans: The clustering algorithm that uses Euclidean distance as the metric to assign the data points to the cluster with nearest centroid.

Must-Link Constraints: Two points that must be assigned to the same cluster.

Semi-Supervised Clustering: An extension of unsupervised clustering algorithms that can utilize the limited supervision given as labeled data or pairwise constraints to improve the clustering results.

Learning Temporal Information from Text

Feng Pan
University of Southern California, USA

INTRODUCTION

As an essential dimension of our information space, *time* plays a very important role in every aspect of our lives. Temporal information is necessarily required in many applications, such as temporal constraint modeling in intelligent agents (Hritcu and Buraga, 2005), semantic web services (Pan and Hobbs, 2004), temporal content modeling and annotation for semantic video retrieval (QasemiZadeh et al., 2006), geographic information science (Agarwal, 2005), data integration of historical stock price databases (Zhu et al., 2004), ubiquitous and pervasive systems for modeling the time dimension of the context (Chen et al., 2004), and so on.

Extracting temporal information from text is especially useful for increasing the temporal awareness for different natural language applications, such as question answering, information retrieval, and summarization. For example, in summarizing a story in terms of a timeline, a system may have to extract and chronologically order events in which a particular person participated. In answering a question as to a person's current occupation, a system may have to selectively determine which of several occupations reported for that person is the most recently reported one (Mani et al., 2004).

This chapter presents recent advances in applying machine learning and data mining approaches to automatically extract explicit and implicit temporal information from natural language text. The extracted temporal information includes, for example, events, temporal expressions, temporal relations, (vague) event durations, event anchoring, and event orderings.

BACKGROUND

Representing and reasoning about temporal information has been a very active area in artificial intelligence and natural language processing. In general there are two approaches to temporal ontology for natural language. The first one is the descriptive approach (Moens and Steedman, 1988; Smith, 1991), which most concerns the descriptive properties of tense and aspect in natural language. The logical and computational approach (Allen and Ferguson, 1997; Hobbs and Pan, 2004) is the other approach that tries to formalize this quite complex ontology, for example, in first-order logic.

More recently, there has been much work on automatically extracting temporal information from text, especially on recognizing events, temporal expressions and relations (Boguraev and Ando, 2005; Mani et al., 2006; Ahn et al., 2007; Chambers et al., 2007). Pan et al. (2006, 2007) shows that implicit and vague temporal information (e.g., vague event durations) is also useful for temporal reasoning and can be learned by standard supervised machine learning techniques (e.g., Support Vector Machines (SVM), Naïve Bayes, and Decision Trees). The TimeBank corpus annotated in TimeML (Pustejovsky et al., 2003) has become a major resource for providing the annotated data for learning temporal information from text. TimeML is a rich specification language for event and temporal expressions in natural language text; unlike most previous attempts, it separates the representation of event and temporal expressions from the anchoring or ordering dependencies that may exist in a given text.

MAIN FOCUS

Machine learning approaches have become more and more popular in extracting temporal information from text. This section focuses on the most recent efforts on extracting both explicit and implicit temporal information from natural language text.

Learning Explicit Temporal Information

The TimeBank corpus currently contains only 186 documents (64,077 words of text), which is a very small size corpus for machine learning tasks. In order to address this challenge of data sparseness, especially for lacking temporal relation annotations, Mani et al. (2006) proposed to use temporal reasoning as an oversampling method. It takes known temporal relations in a document and derives new implied relations from them. There are a total of 745 derivation rules created based on Allen's interval algebra (Allen, 1984). For example, if it is known from a document that event A is before event B and event B includes event C (i.e., event C occurs during event B), then we can derive a new relation that event A is before event C. This oversampling method dramatically expands the amount of training data for learning temporal relations from text. As a result, they have achieved a predictive accuracy on recognizing temporal relations as high as 93% by using a classic Maximum Entropy classifier.

Boguraev and Ando (2005) proposed another approach to the data sparseness problem of the current TimeBank corpus. They developed a hybrid system that combines a finite-state system with a machine learning component capable of effectively using large amounts of unannotated data for classifying events and temporal relations. Specifically, temporal expressions are recognized by the finite-state system; events and temporal relations, especially relations between events and temporal expressions, are learned by the machine learning component with features extracted from the local context and the finite-state system outputs.

Modeling and Learning Implicit Temporal Information

Compared with much work of learning *explicit* temporal information, there has been little work on how to model and extract *implicit* temporal information from natural language text. For example, consider the sentence from a news article:

George W. Bush met with Vladimir Putin in Moscow.

How long did the meeting last? Our first inclination is to say we have no idea. But in fact we do have some idea. We know the meeting lasted more than ten seconds and less than one year. As we guess narrower and narrower bounds, our chances of being correct go down. As part of our commonsense knowledge, we can estimate roughly how long events of different types last and roughly how long situations of various sorts persist. For example, we know government policies typically last somewhere between one and ten years, while weather conditions fairly reliably persist between three hours and one day. There is much temporal information in text that has hitherto been largely unexploited, implicitly encoded in the descriptions of events and relying on our knowledge of the range of usual durations of types of events.

Pan et al. (2006, 2007) presented their work on how to model and automatically extract this kind of implicit and vague temporal information from text. Missing durations is one of the most common sources of incomplete information for temporal reasoning in natural language applications, because very often explicit duration information (e.g., "a five-day meeting", "I have lived here for three years") is missing in natural language texts. Thus, this work can be very important in applications in which the time course of events is to be extracted from text. For example, whether two events overlap or are in sequence often depends very much on their durations. If a war started yesterday, we can be pretty sure it is still going on today. If a hurricane started last year, we can be sure it is over by now.

They have added their new annotations of the implicit and vague temporal information to the TimeBank corpus, and proposed to use normal distributions to model the judgments (i.e., annotations) that are intervals on a scale. The inter-annotator agreement between annotations is defined as the overlapping area between normal distributions. In their corpus every event to be annotated was already annotated in the TimeBank corpus, and annotators were instructed to provide lower and upper bounds on the estimated duration of the event excluding anomalous cases, and taking the entire context of the article into account. A logarithmic scale is used for the annotated data because of the intuition that the difference between 1 second and 20 seconds is significant, while the difference between 1 year 1 second and 1 year 20 seconds is negligible. A preliminary exercise in annotation revealed about a dozen classes of systematic discrepancies among annotators' judgments. So they developed annotation guidelines to make annotators aware of these cases and to guide them in making the judgments.

Then, they applied supervised machine learning techniques (e.g., Support Vector Machines (SVM) with a linear kernel, Naïve Bayes, and Decision Trees C4.5) to the annotated data, and showed that the learning algorithm with the best performance (i.e., SVM) considerably outperformed a baseline and approached human performance. The features they have explored include the local context (i.e., the event word and *n* tokens to its right and left), syntactic relations (i.e., the subject and the object of the event), and WordNet hypernyms (Miller, 1990) of the event word and its subject and object. The features used are very easy to extract and typical in many natural language learning applications.

The new annotations of the implicit temporal information have also been integrated into the TimeBank corpus to enrich the expressiveness of TimeML and also to provide applications that exploit the corpus with the additional implicit information for their temporal reasoning tasks.

FUTURE TRENDS

Given the small size of the current available corpus of temporal information (e.g., TimeBank), more unsupervised and semi-supervised machine learning techniques will be explored for learning temporal information from larger corpora, e.g., the Web. Explicit and implicit temporal information will be extracted and combined together to improve the performance of the current temporal reasoning systems. The method of using normal distributions to model implicit and vague temporal information can be extended from time to other kinds of vague but substantive information.

CONCLUSION

In this chapter we focused on recent work on applying machine learning techniques to automatically extract explicit temporal information from text, including recognizing events, temporal expressions, and temporal relations. Different approaches to the data sparseness problem of the TimeBank corpus were described. We have also presented the work of modeling and learning implicit and vague temporal information from text, which had very little study before.

REFERENCES

Agarwal, P. (2005). Ontological considerations in GIScience. *International Journal of Geographical Information Science*, 19, 501-535.

Ahn, D., Rantwijk, J. van, & Rijke, M. de. (2007). A Cascaded Machine Learning Approach to Interpreting Temporal Expressions, In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics and Human Language Technology (NAACL-HLT)*.

Allen, J. F. (1984). Towards a general theory of action and time. *Artificial Intelligence,* 23, 123-154.

Allen, J. F., & Ferguson, G. (1997). Actions and events in interval temporal logic. In *Spatial and Temporal Reasoning*. O. Stock, ed., Kluwer, Dordrecht, Netherlands, 205-245.

Boguraev, B., & Ando, R. K. (2005). TimeML-Compliant Text Analysis for Temporal Reasoning. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*.

Chambers, N., Wang, S., & Jurafsky, D. (2007). Classifying Temporal Relations Between Events. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics (ACL)*, Prague.

Chen, H., Perich, F., Chakraborty, D., Finin, T., & Joshi, A. (2004). Intelligent agents meet semantic web in a smart meeting room. In *Proceedings of the third International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS)*.

Hobbs, J. R., & Pan, F. (2004). An Ontology of Time for the Semantic Web. *ACM Transactions on Asian Language Processing (TALIP)*, 3(1), 66-85.

Hritcu, C., & Buraga, S. C. (2005). A reference implementation of ADF (Agent Developing Framework): semantic Web-based agent communication. In *Proceedings of the Seventh International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*.

Mani, I., Pustejovsky, J., & Sundheim, B. (2004). Introduction: special issue on temporal information processing. *ACM Transactions Asian Language Information Processing (TALIP)*, 3(1), 1–10.

Mani, I., Verhagen, M., Wellner, B., Lee, C. M., & Pustejovsky, J. (2006). Machine Learning of Temporal Relations. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL)*, Sydney, Australia.

Miller, G. A. (1990). WordNet: an On-line Lexical Database. *International Journal of Lexicography*, 3(4).

Moens, M., & Steedman, M. (1988). Temporal Ontology and Temporal Reference. *Computational Linguistics*, 14(2): 15-28.

Pan, F. & Hobbs, J. R. (2004). Time in OWL-S. In *Proceedings of AAAI Spring Symposium on Semantic Web Services*, Stanford University, CA.

Pan, F., Mulkar, R., & Hobbs, J. R. (2006). Learning Event Durations from Event Descriptions. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL)*, Sydney, Australia, 393-400.

Pan, F., Mulkar-Mehta, R., & Hobbs, J. R. (2007). Modeling and Learning Vague Event Durations for Temporal Reasoning. In *Proceedings of the Twenty-Second Conference on Artificial Intelligence (AAAI)*, Nectar Track, Vancouver, Canada, 1659-1662.

Pustejovsky, J., Hanks, P., Saurí, R., See, A, Gaizauskas, R., Setzer, A., Radev, D., Sundheim, B., Day, D., Ferro, L., & Lazo, M. (2003). The timebank corpus. In *Corpus Linguistics*, Lancaster, U.K.

QasemiZadeh, B., Haghi, H. R., & Kangavari, M. (2006). A Framework for Temporal Content Modeling of Video Data Using an Ontological Infrastructure. In *Proceedings of the Second International Conference on Semantics, Knowledge, and Grid (SKG)*.

Smith, C. S. (1991). *The Parameter of Aspect*. Kluwer Academic Press, Dordrecht.

Zhu, H., Madnick, S.E., & Siegel, M.D. (2004). Effective Data Integration in the Presence of Temporal Semantic Conflicts, In *Proceedings of 11th International Symposium on Temporal Representation and Reasoning (TIME)*, Normandie, France, 109-114.

KEY TERMS

Corpus: A large set of texts that are usually used to do statistical analysis, checking occurrences or validating linguistic rules.

Data Sparseness: Not having enough data for learning the desired patterns or functions.

Finite-State System: A system of behavior modeling that consists of a finite number of states and transitions between those states.

Natural Language: A language that is spoken, written, or signed (visually or tactilely) by humans for general-purpose communications, and natural language processing (NLP) is a field that studies the problems of automatically processing, understanding, and generating the natural language.

Ontology: A data model that represents categories or concepts and relationships between them for a specific domain.

Oversampling: The process of increasing the number of instances in a minority class.

Semi-Supervised Machine Learning: Learning from both labeled and unlabeled training data, usually with a small amount of labeled data and a large amount of unlabeled data.

Supervised Machine Learning: Learning from labeled training data that contains examples with a set of features and desired outputs.

Unsupervised Machine Learning: Learning from unlabeled data that doesn't contain desired outputs.

Learning with Partial Supervision

Abdelhamid Bouchachia
University of Klagenfurt, Austria

INTRODUCTION

Recently the field of machine learning, pattern recognition, and data mining has witnessed a new research stream that is *learning with partial supervision* -LPS- (known also as *semi-supervised learning)*. This learning scheme is motivated by the fact that the process of acquiring the labeling information of data could be quite costly and sometimes prone to mislabeling. The general spectrum of learning from data is envisioned in Figure 1. As shown, in many situations, the data is neither perfectly nor completely labeled.

LPS aims at using available labeled samples in order to guide the process of building classification and clustering machineries and help boost their accuracy. Basically, LPS is a combination of two learning paradigms: supervised and unsupervised where the former deals exclusively with labeled data and the latter is concerned with unlabeled data. Hence, the following questions:

- Can we improve supervised learning with unlabeled data?
- Can we guide unsupervised learning by incorporating few labeled samples?

Typical LPS applications are medical diagnosis (Bouchachia & Pedrycz, 2006a), facial expression recognition (Cohen et al., 2004), text classification (Nigam et al., 2000), protein classification (Weston et al., 2003), and several natural language processing applications such as word sense disambiguation (Niu et al., 2005), and text chunking (Ando & Zhangz, 2005).

Because LPS is still a young but active research field, it lacks a survey outlining the existing approaches and research trends. In this chapter, we will take a step towards an overview. We will discuss (i) the background of LPS, (iii) the main focus of our LPS research and explain the underlying assumptions behind LPS, and (iv) future directions and challenges of LPS research.

BACKGROUND

LPS is about devising algorithms that combine labeled and unlabeled data in a symbiotic way in order to boost classification accuracy. The scenario is portrayed in Fig.2 showing that the combination can mainly be done in two ways: active/passive pre-labeling, or via 'pure' LPS (Fig. 4). We try to draw a clear picture about these schemes by means of an up-to-date taxonomy of methods.

Active and Passive Pre-Labeling

Pre-labeling aims at assigning a label to unlabeled samples (called queries). These samples are used together with the originally labeled samples to train a fully supervised classifier (Fig. 3). "Passive" pre-label-

Figure 1. Learning from data spectrum

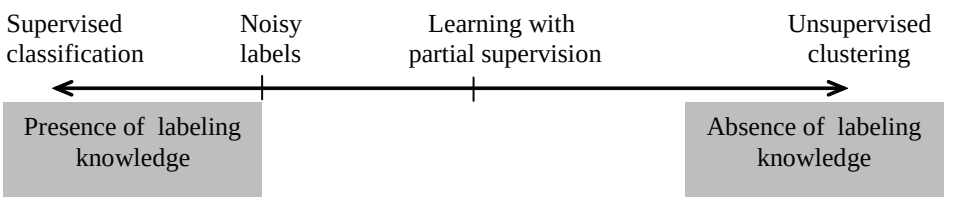


Figure 2. Combining labeled and unlabeled data

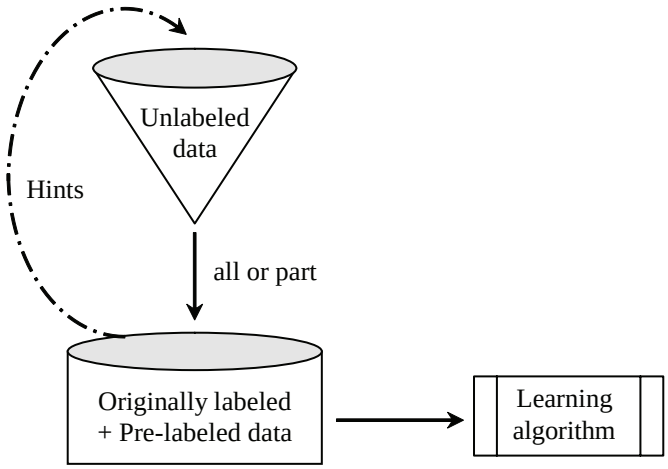


Figure 3. Pre-labeling approaches

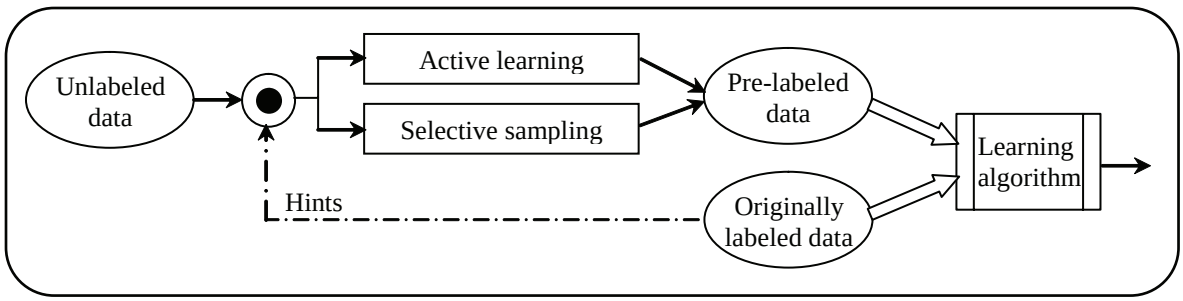
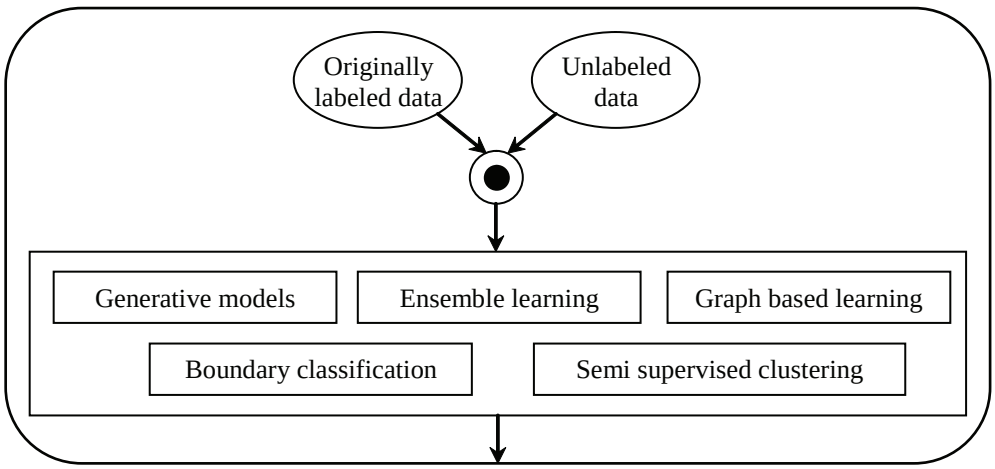


Figure 4. Combining labeled and unlabeled data



ing means that pre-labeling is done automatically and is referred to as selective sampling or self-training. It has been extensively discussed and consists of first training a classifier before using it to label the unlabeled data (for more details see, Bouchachia, 2007). Various algorithms are used to perform selective sampling, such as multilayer perceptron (Verikas et al., 2001), slef-organizing maps (Dara et al., 2002), and clustering techniques (Bouchachia, 2005a). On the other hand, in active learning, queries are sequentially submitted to an oracle for labeling. Different models have been applied; such as neural networks inversion (Baum, 1991), decision trees (Wiratunga et al., 2003), and query by committee (Freund & Shapire, 1997).

Pure LPS

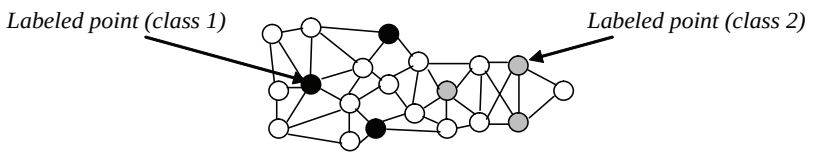
This paradigm corresponds to LPS where the labeled and unlabeled data are reated at the same time. There are several LPS approaches including generative, ensemble learning, graph-based, boundary classification, and semi-supervised clustering models (Fig.4). We briefly outline the concepts behind these approaches without claiming any exhaustiveness due to space limitation.

Much LPS research has focused on *generative approaches* (Nigam et al., 2000)) which seek to optimize a probabilistic model defined by some characteristics such as the cluster center and standard deviation. Usually such models rely on an iterative process to estimate their characteristics. A frequently used method is *Expectation-Maximization* (EM) (Lu *et al.*, 2007).

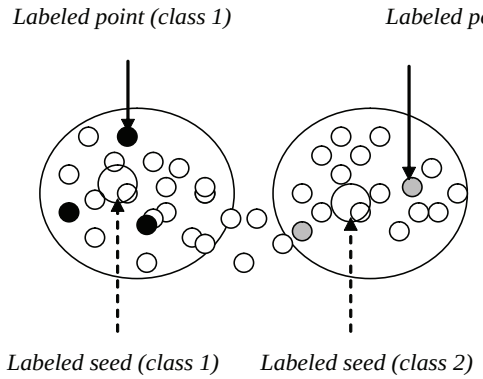
Co-training was introduced in (Blum & Mitchell, 1998). The idea of co-training is to split the features describing the data into two (or more) independent subsets. Each of these is used to train an algorithm. Co-training belongs to the class of *ensemble methods* where individual classifiers are trained and combined to predict the label of data. To this class belong also hybrid approaches like democratic Co-learning (Zhou & Goldman, 2004), and 2v-EM (Ghani, 2002).

Graph-based algorithms consider the problem of learning from hybrid data as a graph problem (Fig.5(a)). The data samples represent the nodes of a graph interconnected by weighted edges. Most of the

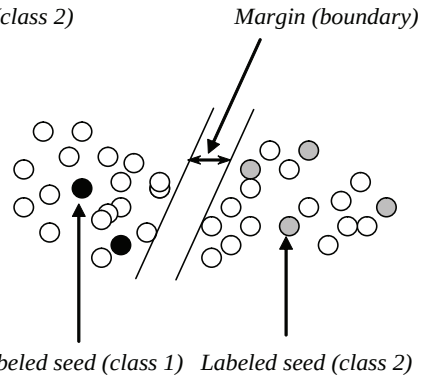
Figure 5. Some LPS methods: (a) graph-based, (b) CPS, (c) Boundary classification (empty circles (unlabeled points))



(a) Data points connected by edges of a graph



(b) Two clusters with two labeled seeds



(c) Two classes separated by a margin

graph-based algorithms are formulated as an objective function consisting of two terms: a loss function and a regularizer. To this category belong many transductive methods such as the *mincut* (Blum et al., 2004), and *random walk-based* (Szummer & Jaakkola, 2002) approaches.

Boundary classification methods (Fig.5(c)), which include *transductive support vector machines* (Yeung & Chang, 2007), *LPS based on Gaussian processes* (Lawrence & Jordan, 2005), and *entropy-based methods* (Jaakkola et al., 1999), rely on the notion of low data density assumption stipulating that the class boundary tends to be of low density.

A slightly different scheme of LPS is *clustering with partial supervision* -CPS- (or s*emi-supervised clustering)* (Fig.5(b))*;* which aims at incorporating some labeled samples before clustering the entire data. The few algorithms developed in this framework are essentially objective function-based including *genetic algorithms* (Demiriz et al., 1999) and *Fuzzy C-Means-based CPS* (Bouchachia & Pedrycz, 2006a). The objective function consists of two terms: one is clustering-oriented and the other is classification-oriented (see next section). Moreover, part of the CPS class are *seed-based* methods (Basu et al., 2002), where the labeled samples are used to generate seeds around which the unlabeled samples are clustered.

MAIN FOCUS

As outlined earlier, in (Bouchachia & Pedrycz, 2006a), a new CPS algorithm has been developed. It is based on an extension of the objective function of Fuzzy C-Means (Bezdek, 1981) to systematically combine labeled and unlabeled data. The objective function consists of two terms. The first term, that is a clustering term, aims at discovering the hidden structure of data. The second term takes the visible structure reflected by the available labels into account. The objective function looks as follows:

$$Objective = Clustering\ Term + \alpha * Classification\ Term$$

where α is a tuning parameter to set the balance between the two terms of the objective function. Higher value indicates more confidence in the labeled data. The outcome of optimizing this function leads to derive the expression of the cluster centers and the membership matrix (membership degree of points to each of the clusters).

This algorithm has shown that by increasing the amount of labeled data, the accuracy of the four proposed approaches increases. Furthermore, unlabeled data does also improve the accuracy. However, the amount of improvement depends on the method used. However, the performance of clustering and classification algorithms is typically influenced by their free parameters. As investigated in (Bouchachia & Pedrycz, 2006b), the effect of distance measures on the performance of CPS is investigated. In this study, the Euclidean distance and three more versatile and adaptive distances: a weighted Euclidean distance, a full adaptive distance, and a kernel-based distance were applied. A comprehensive assessment has shown how the fully adaptive distance allows better performance results but also how the fuzzy C-means based CPS algorithm outperforms other algorithms.

While in most studies, the issue has always been only to check the effect of increasing labeled data on the accuracy of LPS algorithms, it is important to check also the effect of unlabeled data on the accuracy as done in (Bouchachia, 2007). The reason behind the restriction to only the first check is that most of the LPS research is concentrating on semi-supervised classification or transductive learning and therefore, the most important aspect there is the amount of labeled data. On the other hand, the basic assumption that underlies the second check is that the amount of unlabeled data is far bigger than that of labeled data and therefore, the application of clustering techniques should be strongly promoted.

The *cluster assumption* is at the heart of LPS, be it classification or clustering. Indeed, it is worth noting that other assumptions such as the *boundary* (or low data density) assumption, *smoothness and continuity* assumption all implicitly based upon or refer to clustering. Consider the following definitions:

- **Cluster assumption:** Samples lying in the same cluster tend to have the same label.
- **Smoothness assumption:** Samples which are closer to each other in the input space, are likely to be mapped onto the same label in the output space (Zhu et al., 2003).
- **Boundary assumption:** Classes are represented as high density regions separated by valleys (low density regions) (Chapelle & Zien, 2005).

It is easy to see that all of these assumptions convey indirectly the same meaning. Assume that a class consists of many clusters (Bouchachia & Pedrycz, 2006a), then both assumptions: smoothness and low-density separation are taken into account. Nevertheless, these assumptions are seen differently depending on the computational model used in the algorithms; hence, our motivation for approaching the LPS problem from a clustering perspective.

However, despite the well-defined principles and assumptions behind the LPS algorithms, it is still hard to cope with the problem of learning from hybrid data when the amount of labeled data is very small. The reason for that is simple. Few labeled points cannot reflect the distribution of the classes in the input space. This complexity of the task becomes severe if some classes are unknown. A further problem lies in the absence of a thorough evaluation of all available algorithms that would enable the practitioners of data mining and pattern recognition to make the best choice among the available algorithms.

At the current stage of investigation of LPS, one can formulate some key questions:

Question 1: *Is it possible to make a definite choice among all available algorithms?*

Answers 1: As mentioned earlier, there is no single algorithm that can be preferred. The choice can only be motivated by various factors:

- Familiarity of the practitioner with the computational models used by the LPS algorithms, be it probabilistic, graphical, boundary–based or pre-labeling
- Existence of some comparative studies (which, if they exist, are at the current state of affairs are limited to one computational model)
- Sufficient knowledge about the structure of the data (statistical properties). It is important to note that in many approaches, there are strong statistical assumptions about the data that are not true (Cozman & Cohen, 2002)
- Availability of reasonable amount of labeled data reflecting the different regions in each class
- The number of free parameters of each LPS algorithm. The higher the number of free parameters, the higher the tuning effort and the risks of misclassification

Question 2: *Do unlabeled data increase the accuracy of LPS algorithms?*

Answers 2: Most of the authors report that unlabeled data have a positive effect on the accuracy of the LPS algorithms. However, in formal probabilistic terms (Seeger, 2001), many LPS algorithms (e.g., generative and graph-based) show the utility of unlabeled data only if the probability p(x) of unlabeled data does influence the computation of the posterior probability p(y|x). Most of us expect somehow that an improvement, be it minor, in the accuracy as the size of labeled and unlabeled data is increased, since the more the data, the smaller the variance of estimates, and the smaller the classification error. However, this might not be completely true as shown in several studies (Cozman & Cohen, 2002).

Question 3: *Do labeled data increase the accuracy of LPS algorithms?*

Answers 3: Similar to Answer 2, one can a-priori expect that increasing the size of labeled data increases the accuracy. This has been outlined in most LPS studies. Castelli and Cover (1995) have shown that the labeled data contribute exponentially faster than unlabeled data to the reduction of the classification error under some assumptions. However, it has been shown in at least one study (Schohn & Cohn, 2000) that the misclassification error in an active learning setting increases as the size of the labeled data exceeds a certain threshold. To our knowledge, this is the only study that shows that labeled data can harm.

FUTURE TRENDS

It is worth mentioning that LPS is a challenging task. Despite the large variety of LPS algorithms, there are still many open questions. Several axes for future work can be suggested for further investigation:

- Full assessment of the plethora of already proposed algorithms
- Development of further LPS algorithms is expected, advisable and desirable
- Hybrid algorithms are very likely to be investigated. Such algorithms usually tend to avoid making strong assumptions about the models and data

- Semi-automatic LPS algorithms will have a larger share of the algorithms. Semi-automatic here means involving experts in labeling further samples to provide a clearer tendency of the distribution of the labeled data. This refers to active learning. There exist, but very few, approaches relying on active learning to deal with partly labeled data
- Ensemble clusterers and classifiers are widely investigated in the machine learning literature and have proven to outperform individual clustering and classification algorithms. It is, therefore, interesting to consider *ensemble LPS algorithms*. Some initial work in this direction can be found in (D'Alchè-Buc, 2002; Bouchachia, 2007)

CONCLUSION

LPS is very relevant for many applications in the domain of data mining, pattern recognition, and machine learning. LPS has witnessed in recent years much attention from the research community. Its impact will be crucial on the future of such applications since its success reduces the effort and cost invested on labeling. This is especially relevant to particular applications (e.g., chemical, physics, text and data mining) in which labeling is very expensive. However, LPS remains a challenging, but also an exciting, research domain. More investigations are expected to be carried out as outlined in the previous section.

REFERENCES

Ando, R. & Zhangz, T. (2005). A high-performance semi-supervised learning method for text chunking. *Proceedings of the 43rd Annual Meeting of ACL*, pages 1–9.

Bouchachia, A. (2007). Learning with partially labeled data. *Neural Computation and Applications. In press, DOI* 10.1007/s00521-007-0091-0.

Bouchachia, A., & W. Pedrycz, W. (2006a). Data clustering with partial supervision. *Data Mining and Knowledge Discovery*, 12(1):47–78.

Bouchachia, A., & Pedrycz, W. (2006b). Enhancement of fuzzy clustering by mechanisms of partial supervision. *Fuzzy Sets and Systems*, 257(13), pages *1733-1759*.

Bouchachia, A. (2005a). RBF networks for learning from partially labeled data. *Workshop on learning with partially classified training data at the* 22nd *ICML*, pages 10–18.

Bouchachia, A. (2005b). Learning with hybrid data. *The* 5*th IEEE Conference on Intelligent Hybrid Systems*, pages 193-198.

Basu, S., Banerjee, A., & Mooney, R. (2002). Semi-supervised Clustering by Seeding. *The 19th ICML*, pages 19–26.

Baum, E. (1991). Neural net algorithms that learn in polynomial time from examples and queries. *IEEE Transactions on Neural Networks*, 2(1):5–19.

Bezdek J. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum, New York.

Blum, A., Lafferty, J., Rwebangira, M., & Reddy, R. (2004) Semi-supervised Learning using Randomized Mincuts. *The 21thICML*, pages 92–100.

Blum, A., & Mitchell, T. (1998) Combining labeld and unlabaled data with Co-Training. *The 11thConference on Computatioonal Learning Theory*, pages 92–100.

Castelli, V., & Cover, T. M. (1995). On the exponential value of labeled samples. *Pattern Recognition Letters*, *16*, 105–111.

Chapelle, O., & Zien, A. (2005). Semi-supervised classification by low density separation. *The 10thWorkshop on AI and Statistics*, pages 57-64.

Cohen, I., Cozman, F., Sebe, N., Cirelo, M., & Huang, T. (2004). Semi-supervised learning of classifiers: Theory, algorithms, and their application to human-computer interaction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(12), pp: 1553-1567

Cozman, F., & Cohen, I. (2002). Unlabeled data can degrade classification performance of generative classifiers. *International Conference of the Florida Artificial Intelligence Research Society*, pages 327–331.

Dara, R. and Kremer, S., & Stacey, D. (2002). Clustering unlabeled data with SOMs improves classification of labeled real-world data. *World Congress on Computational Intelligence*, pages 2237–2242.

D'Alchè-Buc F., Grandvalet, Y., & Ambroise C.. (2002). Semi-supervised marginboost. *NIPS*,14.

Demiriz, A., Bennett, K., & Embrechts, M. (1999). Semi-supervised clustering using genetic algorithms. *Intelligent Engineering Systems Through ANN*, pages 809–814.

Freund, Y., & Shapire, R. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139.

Ghani, R. (2002). Combining Labeled and Unlabeled Data for Multi-Class Text Categorization. *The 19th ICML*, pages 187-194.

Jaakkola, T., Meila, M., & Jebara, T. (1999). Maximum entropy discrimination. *NIPS*, 12:470-476.

Lawrence, D., & Jordan, I. (2005). Semi-supervised learning via Gaussian processes. *NIPS*, 17: 753-760.

Lu, Y., Tian, Q., Liu, F., Sanchez, M., & Wang, Y. (2007). Interactive semi-supervised learning for micro-array analysis. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 4(2): 190-203

Muslea, I., Minton, S., & Knoblock, C. (2000). Selective sampling with redundant views. *The AAAI conference*, pages 621 – 626.

Nigam, K., McCallum, A., Thrun, S., & Mitchell, T. (2000). Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103–134.

Niu, Z., Ji, D., & Tan, C. (2005). Word sense disambiguation using label propagation based semi-supervised learning. *The 43rd Meeting of ACL*, pages 395–402.

Schohn, G., & Cohn, D. (2000). Less is more: Active learning with support vector machines. *The 17th International Conference on Machine Learning*, pages: 839--846.

Seeger, M. (2001). Learning with labeled and unlabeled data. *Technical report*, Institute for Adaptive and Neural Computation, University of Edinburgh.

Szummer, M. & Jaakkola, T. (2002). Information Regularization with Partially Labeled Data. *NIPS*, 15:1025–1032.

Verikas, A., Gelzinis, A., & Malmqvist, K. (2001). Using unlabeled data to train a multilayer perceptron. *Neural Processing Letters*, 14:179–201.

Weston, J., Leslie, C., Zhou, D., & Noble, W. (2003). Semi-supervised protein classification using cluster kernels. *NIPS*, 16:595-602.

Wiratunga, N., Craw, S., & Massie, S. (2003). Index driven selective sampling for CBR. *The* 5th *International Conference on Case-based Reasoning*, pages 57–62.

Yeung, D. & Chang, H. (2007). A kernel approach for semi-supervised metric learning. *IEEE Transactions on Neural Networks*, 18(1):141-149.

Zhou, Y., & Goldman, S. (2004). Democratic Co-Learning. *The* 16th *International Conference on Tools with Artificial Intelligence*, pages 1082–3409.

Zhu, X., Ghahramani, Z., & Lafferty, J. (2003). Semi-supervised learning using Gaussian fields and harmonic functions. *The 20th ICML*, pages: 912-919.

KEY TERMS

Classification: The process of assigning samples to known classes. In classification, each sample comes in the form of a pair (feature vector, label). If samples are continuous (i.e. signal), the process is known as regression.

Cluster Assumption: The assumption that samples lying in the same cluster tend to have the same label.

Clustering: The process of assigning unlabeled samples to clusters using some similarity measure. Ideally, two criteria have to be satisfied, intra-cluster similarity and inter-cluster dissimilarity.

Induction: The process of inducing a general decision function valid for the whole underlying data set of a learning task.

Semi-Supervised Classification: The process of training a classifier using labeled and unlabeled data, where the unlabeled samples are used to help discover the hidden structure of data and boost the classification accuracy.

Semi-Supervised Clustering: The process of clustering labeled and unlabeled data, where the labeled samples are used to guide the process of grouping and to boost the accuracy of unsupervised clustering.

Semi-Supervised Learning: The process of combining labeled and unlabeled data in order to build a learning system. Usually, the amount of labeled samples is small, while that of unlabeled ones is large.

Transduction: The process of decision making on a particular test set where the aim is to minimize misclassification of those particular points. From the perspective of LPS, the testing set coincides with the set of unlabeled points.

Legal and Technical Issues of Privacy Preservation in Data Mining

Kirsten Wahlstrom
University of South Australia, Australia

John F. Roddick
Flinders University, Australia

Rick Sarre
University of South Australia, Australia

Vladimir Estivill-Castro
Griffith University, Australia

Denise de Vries
Flinders University, Australia

INTRODUCTION

To paraphrase Winograd (1992), we bring to our communities a tacit comprehension of right and wrong that makes social responsibility an intrinsic part of our culture. Our **ethics** are the moral principles we use to assert social responsibility and to perpetuate safe and just societies. Moreover, the introduction of new technologies can have a profound effect on our ethical principles. The emergence of very large databases, and the associated automated data analysis tools, present yet another set of ethical challenges to consider.

Socio-ethical issues have been identified as pertinent to data mining and there is a growing concern regarding the (ab)use of sensitive information (Clarke, 1999; Clifton et al., 2002; Clifton and Estivill-Castro, 2002; Gehrke, 2002). Estivill-Castro et al., discuss surveys regarding public opinion on personal privacy that show a raised level of concern about the use of private information (Estivill-Castro et al., 1999). There is some justification for this concern; a 2001 survey in InfoWeek found that over 20% of companies store customer data with information about medical profile and/or customer demographics with salary and credit information, and over 15% store information about customers' legal histories.

BACKGROUND

Data mining itself is not ethically problematic. The ethical and legal dilemmas arise when mining is executed over data of a personal nature. Perhaps the most immediately apparent of these is the invasion of privacy. Complete **privacy** is not an inherent part of any society because participation in a society necessitates communication and negotiation, which renders absolute privacy unattainable. Hence, individual members of a society develop an independent and unique perception of their own privacy. Privacy therefore exists within a society only because it exists as a perception of the society's members. This perception is crucial as it partly determines whether, and to what extent, a person's privacy has been violated.

An individual can maintain their **privacy** by limiting their accessibility to others. In some contexts, this is best achieved by restricting access to personal information. If a person considers the type and amount of information known about them to be inappropriate, then they perceive their privacy to be at risk. Thus, privacy can be violated when information concerning an individual is obtained, used, or disseminated, especially if this occurs without their knowledge or consent.

Huge volumes of detailed personal data are regularly collected and analysed by marketing applications (Fienberg, S. E. 2006; Berry and Linoff, 1997), in

which individuals may be unaware of the behind-the-scenes use of data, are now well documented (John, 1999). However, privacy advocates face opposition in their push for legislation restricting the secondary use of personal data, since analysing such data brings collective benefit in many contexts. DMKD has been instrumental in many scientific areas such as biological and climate-change research and is also being used in other domains where privacy issues are relegated in the light of perceptions of a common good. These include genome research (qv. (Tavani, 2004)), combating tax evasion and aiding in criminal investigations (Berry and Linoff, 1997) and in medicine (Roddick et al., 2003).

As **privacy** is a matter of individual perception, an infallible and universal solution to this dichotomy is infeasible. However, there are measures that can be undertaken to enhance privacy protection. Commonly, an individual must adopt a proactive and assertive attitude in order to maintain their privacy, usually having to initiate communication with the holders of their data to apply any restrictions they consider appropriate. For the most part, individuals are unaware of the extent of the personal information stored by governments and private corporations. It is only when things go wrong that individuals exercise their rights to obtain this information and seek to excise or correct it.

MAIN FOCUS

Data Accuracy

Mining applications involve vast amounts of data, which are likely to have originated from diverse, possibly external, sources. Thus the quality of the data cannot be assured. Moreover, although data pre-processing is undertaken before the execution of a mining application to improve data quality, people conduct transactions in an unpredictable manner, which can cause personal data to expire. When mining is executed over expired data inaccurate patterns are more likely to be revealed.

Likewise, there is a great likelihood of errors caused by mining over poor quality data. This increases the threat to the data subject and the costs associated with the identification and correction of the inaccuracies. The fact that data are often collected without a preconceived hypothesis shows that the data analysis used in DMKD are more likely to be **exploratory** (as opposed to the **confirmatory analysis** exemplified by many statistical techniques). This immediately implies that results from DMKD algorithms require further confirmation and/or validation. There is a serious danger of inaccuracies that cannot be attributed to the algorithms *per se*, but to their exploratory nature.

This has caused some debate amongst the DMKD community itself. Freitas (2000) has argued that mining association rules is a deterministic problem that is directly dependent on the input set of transactions and thus association rules are inappropriate for prediction, as would be the case of learning classifiers. However, most uses of association rule mining are for extrapolation to the future, rather than characterisation of the past.

The sharing of corporate data may be beneficial to organisations in a relationship but allowing full access to a database for mining may have detrimental results. The adequacy of traditional database security controls are suspect because of the nature of inference. Private and confidential information can be inferred from public information.

The following measures have thus been suggested to prevent unauthorised mining:

- **Limiting access to the data.** By preventing users from obtaining a sufficient amount of data, consequent mining may result in low confidence levels. This also includes query restriction, which attempts to detect when compromise is possible through the combination of queries (Miller and Seberry, 1989).
- **Anonymisation.** Any identifying attributes are removed from the source dataset. A variation on this can be a filter applied to the ruleset to suppress rules containing identifying attributes.
- Dynamic **Sampling**. Reducing the size of the available data set by selecting a different set of source tuples for each query.
- **Authority control** and cryptographic techniques. Such techniques effectively hide data from unauthorised access but allow inappropriate use by authorised (or naive) users (Pinkas, 2002).
- **Data perturbation**. Altering the data, by forcing aggregation or slightly altering data values, useful mining may be prevented while still enabling the planned use of the data. Agrawal and Srikant (2000) explored the feasibility of privacy-preservation by using techniques to perturb sensitive values in data.

- **Data swapping.** Attribute values are interchanged in a way that maintains the results of statistical queries (Evfimievski et al., 2002).
- **The elimination of unnecessary groupings.** By assigning unique identifiers randomly; they serve only as unique identifiers. This prevents meaningful groupings based on these identifiers yet does not detract from their intended purpose.
- **Data augmentation**. By adding to the data in non-obvious ways, without altering their usefulness, reconstruction of original data can be prevented.
- **Alerting.** Labelling potentially sensitive attributes and attribute values and from this calculating an estimate of the sensitivity of a rule (Fule and Roddick, 2004).
- **Auditing.** The use of auditing does not enforce controls, but could detect misuse so that appropriate action may be taken.

Issues relating to the computational cost of privacy preservation are discussed by Agrawal et al. (2004).

Legal Liability

When personal data have been collected it is generally decontextualised and separated from the individual, improving privacy but making misuse and mistakes more likely (Gammack and Goulding, 1999). Recently, there has been a trend to treat personal data as a resource and offer it for sale. Information is easy to copy and re-sell. The phrase *data mining* uses the metaphor of the exploitation of natural resources, further contributing to the perception of data as commodity. Moreover, the question of whether it is appropriate in terms of human rights to trade in personal data has seen insufficient academic and legal debate. The negative consequences of such trade are similar to those of data mining: transgression of privacy and the negative impacts of inaccurate data. However, the repercussions of inaccurate data are more serious for organisations trading in personal data, as the possibility of legal liability is introduced. There is the potential for those practising data trade or data mining to make mistakes and as a consequence lose heavily in the courts.

Compensation may be ordered against any organisation that is found to have harmed (or failed to prevent harm to) an individual to whom it owed a duty of care. Once liability (the tort of negligence) has been established, the plaintiff can claim financial compensation for any consequential losses caused by the negligent act (Samuelson, 1993). The extent and exact nature of the losses is, for the most part, unique to each plaintiff, but the boundaries of negligence are never closed. A mining exercise might erroneously declare an individual a poor credit risk, and decisions may be made prejudicial to that individual on that basis.

In some cases, algorithms may classify correctly, but such classification could be on the basis of controversial (ie. ethically sensitive) attributes such as sex, race, religion or sexual orientation. This could run counter to **Anti-Discrimination legislation**. In some cases, such as artificial neural networks, SVMs and nearest neighbour classifiers, which do not make their knowledge explicit in rules, the use of controversial classification attributes may be hard to identify. Even with methods that make transparent their classification, such as decision trees, there is little to prevent a corporation using rules based on controversial attributes if that improves accuracy of the classification. Individuals who suffer denial of credit or employment on the basis of race, sex, ethnicity or other controversial attributes in a context where this is contrary to law are in a strong position to demonstrate harm *only* if they illustrate the artificial classifiers are using such attributes. The question is how they obtain access to the classifier results.

In the event that the person loses money or reputation as a result of this, courts may award damages. Moreover, since the potential for inaccuracies involved in the exercise is great, it is conceivable that the courts might apply a higher than usual standard of care in considering whether an organisation has breached its duty to a plaintiff sufficiently to amount to **negligence**. Only time will tell.

Another legal issue is whether organisations manipulating personal data can be considered capable of defaming a person whose data they have mined. It is quite conceivable that since mining generates previously unknown information, the organisation using the mining tool can be considered the author of the information for the purposes of defamation law. Moreover, it can be argued that organisations trading in personal data are analogous to publishers, as they are issuing collections of data for sale and distribution. Hence, if the information is capable of being deemed defamatory by the courts, the data mining organisations are capable of being found liable for damages in this tort also. One difficulty is that the terms *author* and *publisher* have

long been associated with text or music, not data. Note that census data also faces this challenge and other technologies are complicating the issue still further. Consider aerial/satellite photography that can now achieve resolution to within a few metres and which can be freely purchased over the Internet. What is the resolution that makes such data be considered personal data? How can individuals living at identifiable houses decide if the aerial photo is to be used for a potential beneficial analysis, such as bush fire risk analyses of their property, or an analysis that could be considered **defamatory** or **discriminatory**?

Market analysts often see privacy concerns as unreasonable. Privacy is an obstacle to understanding customers and to supplying better suited products. Hundreds of millions of personal records are sold annually in the US by 200 **superbureaux** to direct marketers, private individuals, investigators, and government agencies (Laudon, 1996).

We are in urgent need of an extended interpretation of existing tort doctrine, or preferably a broadening of the boundaries of the current doctrines. Indeed, Samuelson (1993) warns that the engineered, technological nature of electronic information dissemination suggests a greater liability for its disseminators. Commonly, the conveyers of information are excused from liability if they are simply the carriers of the information from the publishers to the public—a book store selling a book that carries defamatory material will be excused from liability that might rightly attach to the author and the publisher. It is quite possible that a mining exercise, particularly one that had mined inaccurate data, might be deemed by the courts to be an exercise in publishing, not just in dissemination.

SOLUTIONS AND FUTURE TRENDS

Anonymisation of Data

One solution to the invasion of privacy problem is the **anonymisation** of personal data. This has the effect of providing some level of privacy protection for data subjects. However, this would render obsolete legitimate mining applications that are dependent on identifiable data subjects, and prevent many mining activities altogether. A suggested compromise is the empowerment of individuals to dictate the amount and type of personal data they consider appropriate for an organisation to mine.

While anonymisation of data is a step in the right direction, it is the weakest of the possible options. It is well known that additional information about an individual can easily be used to obtain other attributes. Moreover, grouping two sets of anonymised information can result in disclosure. Identifier removal (such as name, address, phone number and social security number) can be insufficient to ensure privacy (Klosgen, 1995). Anonymisation is a form of cell suppression, a technique applied on statistical databases. Indeed, the research agenda is still far from closed since most of the solutions proposed so far in the DMKD community (Piatetsky-Shapiro, 1995) are easily translated to existing methods for statistical databases.

Data perturbation is thus a promising alternative. Clifton and Marks (1996; 1999) indicated new and renewed threats to privacy from mining technology and Estivill-Castro and Brankovic (1999) indicated the potential of data perturbation methods which was subsequently adopted by Agrawal and Srikant (2000).

Inaccurate Data

The data quality issue is more difficult to resolve. Inaccurate data remains undetected by the individual until he or she experiences some associated repercussion, such as a denial of credit. It is also usually undetected by the organisation, which lacks the personal knowledge necessary for the exposure of inaccuracies. The adoption of data quality management strategies by the organisation, coupled with the expedient correction of inaccuracies reported by individuals and intermittent data cleansing may go some way to resolving the dilemma.

Legal Solutions

Legal regulation of applied technology is currently one of the more pressing needs facing policy-makers. But how does one approach the development of what is needed? Legislative reform may be unsuitable as it is an unwieldy tool in a rapidly expanding technical environment. Common law change is probably a more appropriate and suitable mechanism of legal regulation as it can be creatively applied to novel situations. The disadvantage of the common law, however, is that there needs to be a number of precedent court cases upon which to build common law principles, and litigation in this age of mediated dispute resolution and in con-

fidence settlements is becoming a rare phenomenon. Furthermore, the common law's record on subjects such as the protection of privacy and dealing with the legal implications of applied technologies is weak. This is a rapidly expanding socio-economic landscape, and the law is not keeping pace. Public awareness of legal suits alleging negligence and defamation may possibly have some prophylactic effect upon the potential transgressions of contemporary technology.

Another weakness of legal regulation is the fact that the jurisdictional boundaries that determine the limits of our legal system were decided in ignorance of technological developments that render these boundaries virtually irrelevant, given the international structure of many organisations and the burgeoning international presence of an individual on the Internet. If an Australian were to conduct a transaction and provide data for a multinational organisation registered in Europe via a web site physically situated in the United States, which legal system governs the transaction and its legal repercussions? This is a legal dilemma not lost on international lawyers, and one that does not readily admit of a simple solution.

CONCLUSION

The primary consideration of any future research should be at least maintenance, and preferably enhancement, of ethical flexibility. Solutions reconciling any issues must not only be applicable to the ever-changing technological environment, but also flexible with regard to specific contexts and disputes. Moreover, we must be able to identify ethical dilemmas as they arise and derive solutions in a timely, preferably concurrent manner. Many ethical issues overlap and have effects on each other. We need to identify any commonalities and differences that exist and exploit them to derive solutions that enable us to uphold, or extend, our ethical standards.

In terms of practical issues, the equality of access argument (the technological *haves and have-nots* (Baase 1997)) has not been considered in this chapter. Indeed, data mining may be one context in which the have-nots hold the advantage.

We exist in an environment in which technology has increasing social relevance. The challenge now is to implement methods of assessing the social impact of an emergent technology, providing us with the capacity to use the tools technology provides wisely and with consideration for our culture and its future.

REFERENCES

Agrawal, R., & Srikant, R. (2000). Privacy-preserving data mining, in 'ACM SIGMOD Conference on the Management of Data'. *ACM*, 439–450.

Agrawal, S., & Haritsa, J. (2005). A framework for high-accuracy privacy-preserving mining, in '21st Int. Conf. on Data Engineering'. *IEEE*, 193–204.

Agrawal, S., Krishnan, V., & Haritsa, J. (2004). *On addressing efficiency concerns in privacy preserving data mining*. In Proceedings of the 9th DASFAA, LNCS 2973, Springer, Korea, 113–124.

Baase, S. (1997). *A gift of fire: Social, legal, and ethical issues in computing*. Prentice-Hall.

Banks, D.L., & Said, Y.H. (2006). Data mining in electronic commerce. *Statistical Science 21*(2), 234-246.

Berry, M. & Linoff, G. (1997). Data mining techniques for marketing, sales and customer support. John Wiley.

Clarke, R. (1999). Person-location and person-tracking: Technologies, risks and policy implications. In *Proceedings of the21st International Conference on Privacy and Personal Data Protection*, 131–150.

Clifton, C. (1999). Protecting against data mining through samples. In *Proceedings of the Thirteenth Annual IFIP Working Conference on Database Security.* Seattle.

Clifton, C. & Estivill-Castro, V., eds (2002). In Proceedings of the IEEE Data Mining Workshop on Privacy, Security, and Data Mining, CRPIT 14. Maebashi, Japan.

Clifton, C., et al. (2002). Tools for privacy preserving data mining. *SigKDD Explorations 4*(2), 28–34.

Clifton, C. & Marks, D. (1996). Security and privacy implications of data mining. In *Proceedings of the ACM SIGMOD Workshop on DMKD.* Montreal.

Estivill-Castro, V. & Brankovic, L. (1999). *Data swapping: Balancing privacy against precision in mining*

for logic rules. In Proceedings of the DAWAK'99, Springer LNCS 1676, Florence, 389–398.

Estivill-Castro, V., Brankovic, L., & Dowe, D. (1999). Privacy in data mining. *Privacy, Law and Policy Reporter 6*(3), 33–35.

Evfimievski, A., et al. (2002). Privacy preserving mining of association rules. In *Proceedings of the Eighth SIGKDD International Conference on KDDM, ACM.*

Fienberg, S. E. (2006). Privacy and confidentiality in an e-commerce world: Data mining, data warehousing, matching and disclosure limitation. *Statistical Science 21*(2), 143-154.

Freitas, A. (2000). Understanding the crucial differences between classification and discovery of association rules-a position paper. *SIGKDD Explorations 2*(1), 65–68.

Fule, P. & Roddick, J. (2004). Detecting privacy and ethical sensitivity in data mining results. In *Proceedings of the 27th Australasian Computer Science Conference*, CRPIT 26, ACS, 159–166.

Gammack, J., & Goulding, P. (1999), Ethical responsibility and management of knowledge. *ACJ 31*(3), 72–77.

Gehrke, J., ed. (2002). Special Issue on Privacy and Security. *SigKDD Explorations, 4*.

John, G. (1999). Behind-the-scenes data mining. *SIGKDD Explorations 1*(1), 9–11.

Johnson, D. & Nissenbaum, H. (1995). *Computers, ethics and social values*. Prentice-Hall, New Jersey.

Klosgen, W. (1995). KDD: Public and private concerns. *IEEE Expert 10*(2), 55–57.

Laudon, K. (1996). Markets and privacy. *CACM 39*(9), 92–104.

Miller, M., & Seberry, J. (1989). Relative compromise of statistical databases. *ACJ 21*(2), 56-61.

Piatetsky-Shapiro, G. (1995). Knowledge discovery in personal data vs. privacy: A mini-symposium. *IEEE Expert 10*(2), 46–47.

Pinkas, B. (2002). Cryptographic techniques for privacy-preserving data mining. *SigKDD Explorations 4*(2), 12–19.

Roddick, J., Fule, P., & Graco, W. (2003). Exploratory medical knowledge discovery: Experiences and issues. *SigKDD Explorations 5*(1) 94-99.

Samuelson, P. (1993). Liability for defective electronic information. *CACM 36*(1), 21–26.

Subirana, B. & Bain, M. (2006). Legal programming. *CACM 49*(9), 57-62.

Tavani, H. (2004). Genomic research and data-mining technology: Implications for personal privacy and informed consent. *Ethics and IT 6*(1), 15–28.

Winograd, T. (1992). Computers, ethics and social responsibility. In Proceedings to Johnson and Nissenbaum, (1995), 25–39.

KEY TERMS

Authority control: Various methods of controlling access to data.

Data Anonymisation: The removal of attribute values that would allow a third party to identify the individual.

Data Perturbation: Altering data, by forcing aggregation or slightly altering data values.

Multi Level Security (MLS): Extends conventional security measures by classifying data according to its confidentiality.

Sampling: The selection of a subset of data for analysis.

Spyware: Software that covertly gathers user information.

Statistical Compromise: A situation in which a series of queries may reveal confidential information.

Leveraging Unlabeled Data for Classification

Yinghui Yang
University of California, Davis, USA

Balaji Padmanabhan
University of South Florida, USA

INTRODUCTION

Classification is a form of data analysis that can be used to extract models to predict categorical class labels (Han & Kamber, 2001). Data classification has proven to be very useful in a wide variety of applications. For example, a classification model can be built to categorize bank loan applications as either safe or risky. In order to build a classification model, training data containing multiple independent variables and a dependant variable (class label) is needed. If a data record has a known value for its class label, this data record is termed "labeled". If the value for its class is unknown, it is "unlabeled". There are situations with a large amount of unlabeled data and a small amount of labeled data. Using only labeled data to build classification models can potentially ignore useful information contained in the unlabeled data. Furthermore, unlabeled data can often be much cheaper and more plentiful than labeled data, and so if useful information can be extracted from it that reduces the need for labeled examples, this can be a significant benefit (Balcan & Blum 2005). The default practice is to use only the labeled data to build a classification model and then assign class labels to the unlabeled data. However, when the amount of labeled data is not enough, the classification model built only using the labeled data can be biased and far from accurate. The class labels assigned to the unlabeled data can then be inaccurate.

How to leverage the information contained in the unlabeled data to help improve the accuracy of the classification model is an important research question. There are two streams of research that addresses the challenging issue of how to appropriately use unlabeled data for building classification models. The details are discussed below.

BACKGROUND

Research on handling unlabeled data can be approximately grouped into two streams. These two streams are motivated by two different scenarios.

The first scenario covers applications where the modeler can acquire, but at a cost, the labels corresponding to the unlabeled data. For example, consider the problem of predicting if some video clip has suspicious activity (such as the presence of a "most wanted" fugitive). Vast amounts of video streams exist through surveillance cameras, and at the same time labeling experts exist (in law enforcement and the intelligence agencies). Hence labeling any video stream is possible, but is an expensive task in that it requires human time and interpretation (Yan et al 2003). A similar example is in the "speech-to-text" task of generating automatic transcriptions of speech fragments (Hakkani-Tur et al 2004, Raina et al 2007). It is possible to have people listen to the speech fragments and generate text transcriptions which can be used to label the speech fragments, but it is an expensive task. The fields of active learning (e.g. MacKay (1992), Saar-Tsechansky & Provost (2001)) and optimal experimental design (Atkinson 1996) addresses how modelers can selectively acquire the labels for the problems in this scenario. Active learning acquires labeled data incrementally, using the model learned so far to select particularly helpful additional training examples for labeling. When successful, active learning methods reduce the number of instances that must be labeled to achieve a particular level of accuracy (Saar-Tsechansky & Provost (2001)). Optimal experimental design studies the problem of deciding which subjects to experiment on (e.g. in medical trials) given limited resources (Atkinson 1996).

The second scenario, the focus in this chapter, covers applications where it is not possible to acquire the

unknown labels or such acquisition is not an option. The extreme cases of the previous scenario where the costs are prohibitively high can also be considered in this set. For example, consider the problem of predicting the academic performance (i.e. the graduating GPA) of thousands of current applicants to an undergraduate program. Ample data exists from the performance of ex-students in the program, but it is impossible to "acquire" the graduating GPA of current applicants. In this case is the unlabeled data (i.e. the independent variables of the current applicants) of any use in the process of building a model? A stream of recent research (Blum & Mitchell (1998), Joachims (1999), Chapelle (2003)) addresses this problem and presents various methods for making use of the unlabeled data for this context.

To some extent, approaches used to learning with missing values can be applied to learning the labels of the unlabeled data. One standard approach to learning with missing values is the EM algorithm (Dempster et al. 1977). The biggest drawback of such approaches is that they need to assume the class label follows a certain distribution.

A second approach for this (Blum & Mitchell, 1998) is co-training (and variants (Yarowsky, 1995)) which was initially applied to Web page classification, since labeling Web pages involves human intervention and is expensive. The idea is to first learn multiple classifiers from different sets of features. Each classifier is then used to make predictions on the unlabeled data and these predictions are then treated as part of the training set for the other classifiers. This approach works well for Web page categorization since one classifier can be trained based on words within pages while another (using different features) can be trained on words in hyperlinks to the page. This approach is in contrast with self-training where a classifier uses its own (selected) predictions on the unlabeled data to retrain itself.

Another approach is to use clustering and density estimation to first generate a data model from both the labeled and unlabeled data (e.g. Chapelle, 2003). The labels are then used for labeling entire clusters of data, or estimating class conditional densities which involves labeling of the unlabeled data dependent on their relative placement in the data space with respect to the original labeled data. A popular approach for implementing this idea is using generative mixture models and the EM algorithm. The mixture models are identified using unlabeled data, and then the labeled data is used to determine the classes to assign to the (soft) clusters generated from the combined data.

There is also work on integrating these ideas into a specific classifier, such as the development of Transductive Support Vector Machines (Joachims 1999). Extending the concept of finding optimal boundaries in a traditional SVM, this work develops methods to learn boundaries that avoid going through dense regions of points both in labeled as well as unlabeled data.

To summarize, the prior research on learning with unlabeled data focuses either on selecting unlabeled data to acquire labels, or use models built on labeled data to assign labels to unlabeled data.

MAIN FOCUS OF THE CHAPTER

In this chapter, we focus on an approach for using the unlabeled data in the case where labels cannot be acquired. This approach is different from the ones discussed above in that it does not involve assigning labels to the unlabeled data. Instead, this approach augments the features (independent variables) of the labeled data to capture information in the unlabeled data. This approach is based on the intuition that the combined labeled and unlabeled data can be used to estimate the joint distribution of attributes among the independent variables better, than if this was estimated from the labeled data alone. Specifically, if interactions (or patterns) among variables turn out to be useful features for modeling, such patterns may be better estimated using all available data.

The Approach

The approach and alternatives for comparison are pictorially illustrated in Figure 1. The approach presented is the path on the right (#3). First the column represented by the target attribute (Y) is removed, and the labeled and unlabeled data are combined into one large dataset. Then a pattern discovery procedure (e.g. a procedure to discover frequent itemsets) is applied to learn a set of patterns from this data. Let the number of patterns learned from just the independent variables of both the labeled and unlabeled data be Q_2. Each of these patterns then is used to create binary variables P_1, P_2, … , P_{Q2} indicating whether each given pattern is present in each record of the dataset. For example, for pattern number Q_2, we check whether a data record contains

this pattern. If it does, then the value for the newly created attribute P_{Q2} is 1, otherwise it is 0. A classifier is then built using the labeled data which now has the original variables augmented with features learned from the combined data.

The following example illustrates how the approach works. Assume there are two independent variables (X_1 and X_2) in the data (both labeled and unlabeled). The labeled data has a dependent variable Y, but the unlabeled data doesn't. We take X_1 and X_2 in both the labeled data and unlabeled data and find frequent patterns in them (for example, $\{X_1=2, X_2=4\}$ occurs in many data records, so it's a frequent pattern). Then, a new independent variable $P_1=\{X_1=2, X_2=4\}$ is created to augment the existing ones. If a data record has the pattern $\{X_1=2, X_2=4\}$, then $P_1=1$, otherwise 0. At last, we use X_1, X_2, $\{X_1=2, X_2=4\}$ and Y in the labeled data to build classification models.

The intuition behind this approach is founded on three observations. First, there are often systematic interactions among independent variables in large high dimensional datasets. Second, often a subset of

Figure 1. Pictorial illustration of the approach (#3) and two comparison approaches (#1 and #2)

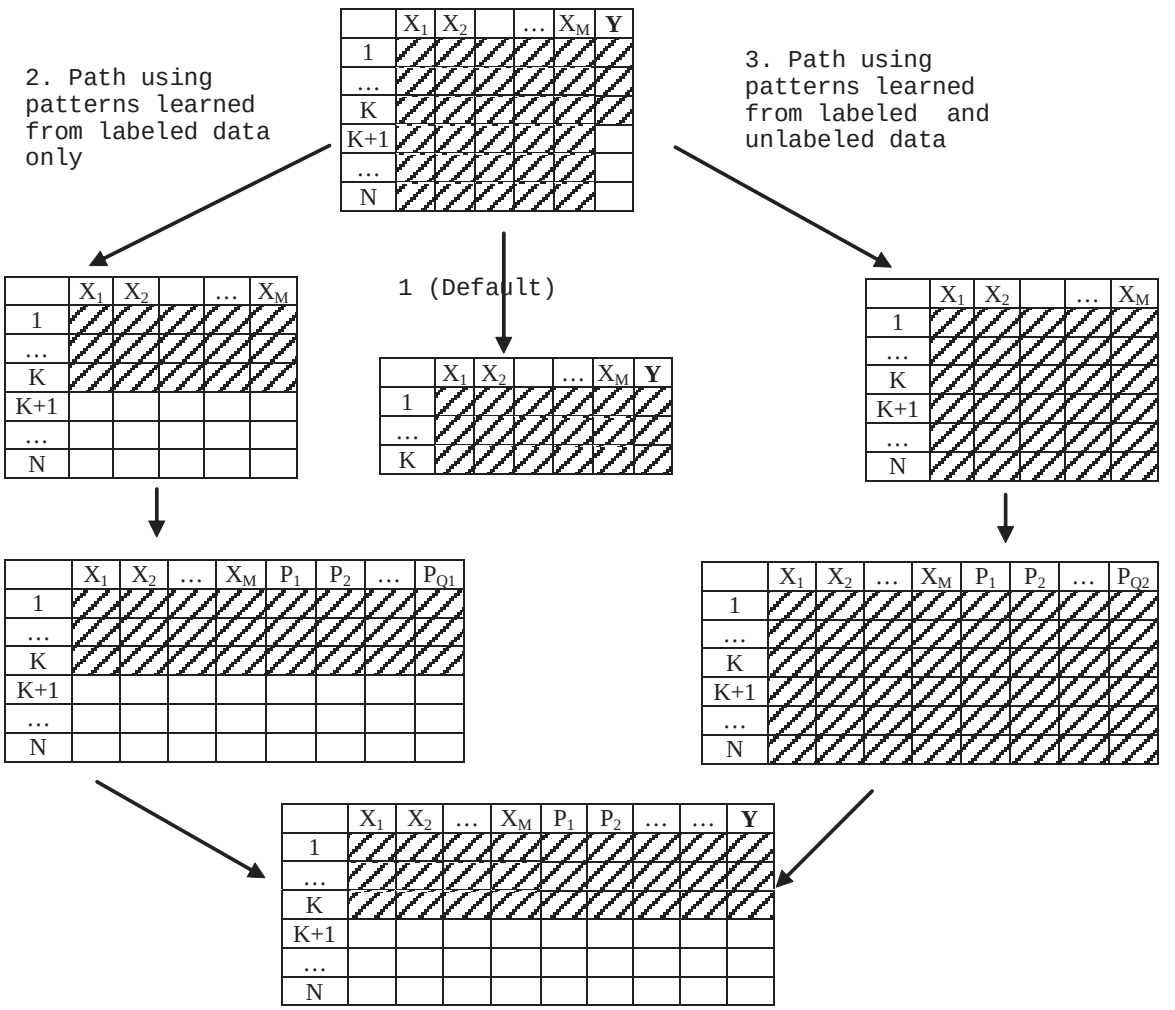


Figure 2. A generic algorithm for using patterns to exploit unlabeled data for classification

Input: Labeled data $D_L = \{t_1, t_2, \ldots, t_k\}$, Unlabeled data $D_{UL} = \{t_{k+1}, t_{k+2}, \ldots, t_N\}$, Pattern Discovery Procedure Γ, Classifier M, Feature Selection Procedure F.

Output: A specific classification model m.

1. $D = D_L \cup D_{UL}$ /* combine unlabeled with labeled data for pattern discovery */
2. Let $P = \Gamma(D)$ /* learn patterns among the explanatory variables */
3. $Q_2 = |P|$ /* index corresponds to path 3 in figure 1 */
4. For all $t_i \in D_L$ {
5. $t_i = t_i \cup \{b_1, b_2, \ldots, b_{Q2}\}$
   where b_j=1 if pattern $p_j \in P$ is present in t_i or b_j=0 otherwise
6. } /* all transactions in labeled data are now augmented with features learned from the combined data */
7. $D_{new} = F(D_L)$
8. $m = M(D_{new})$
9. output m

these patterns is useful in predicting the desired target variable. These two observations actually form the basis for a large body of work on feature construction or feature discovery (Fawcett 1993). Third, given the large search space of interactions, having more data will typically help in learning a more stable (or true) set of patterns. Hence for applications where there is a lot of unlabeled data, learning patterns from the entire data and then building a model is likely to make effective use of all available data. This is based on similar ideas (but not specific to patterns) in the literature that vast unlabeled data is often more likely to be representative of a general true underlying distribution than limited labeled data (Seeger 2001, Peng et al. 2003).

The method is formalized in Figure 2 Steps 1-3 learn the set of patterns using combined data. Steps 4-6 create new binary features in the data corresponding to the patterns learned. Given that this may result in a large number of variables created, step 7 applies a feature selection procedure to select the relevant features before the model is built (step 8).

In order to test if our approach really does exploit the unlabeled data, we highlight alternative approaches in Figure 1. The main comparison should be done with the path on the left (#2), that uses exactly the same process outlined above, except that even the process of learning patterns is only done using only the labeled data – hence this approach also uses patterns for feature construction, but does not use any unlabeled data. Path #1 in the middle is a "default" approach where the labeled data given is directly used to build the model.

To implement the method presented in Section 2 we chose itemsets (Agrawal et al 1995) as the representation for patterns and use standard itemset discovery algorithms to learn the patterns; we also use J4.8, a standard decision tree model in the open source data mining package weka , as the underlying classifier.

We use Web browsing datasets in these experiments. Each dataset consists of a set of sessions from different users mixed together, and the target variable in these is to predict the user corresponding to each session. To generate these datasets we randomly pick and combine (known) user sessions from a master database of user sessions. Note that most Web sites have some user sessions in which they know exactly who the user was (perhaps where the user signed in, made a purchase, or came in with an old cookie), but they also have several (unlabeled) sessions in which user activity is observed without precise identifying information on the user, and hence this particular application also falls under the general type of problems motivated in the introduction.

We compare three ways of building predictive models:

M1: The "default" method, where just the original attributes are used as independent variables.
M2: This is path #2 in Figure 1 where patterns, and features, are generated only from labeled data.
M3: This is path #3 in Figure 1 where both the labeled and unlabeled data were used as described.

The difference between M2 and M3 is that in M3 the frequent patterns are generated from the entire data set, including both labeled and unlabeled data. Hence this difference represents how well the unlabeled data was used. For each dataset we vary the percentage of labeled data assumed to be available to test how the method performs for different amounts of labeled/unlabeled data. The detailed results can be found in Yang & Padmanabhan (2007). On average the improvement is significant in all cases, and the improvement is significant mostly when the amount of labeled data is low. This corresponds well to the observation that when there is enough labeled data to measure the interactions appropriately then unlabeled data is less likely to be useful. Also when the average model accuracy on unlabeled data is high, the improvements are likely to be lower (since all the models perform well), but when the accuracy on unlabeled data is low, the improvement is greater. There are also a few negative numbers for the improvements, and this is certainly possible due to the specific sample, dataset and model built. While these results suggest the potential benefits that can be had in using unlabeled data when the percentage of labeled data is low, certainly further experiments across a larger number of datasets and with multiple runs are needed to get a more complete picture of the benefits and the exact circumstances in which the unlabeled data is useful.

FUTURE TRENDS

The problem of exploiting unlabeled data is relatively new, and is one that can have several applications. For instance, the co-training method was really motivated by the fact that millions of unlabeled Web pages exist and can be used in models. In marketing such situations arise when firms have a database of (potentially) millions of individuals to, say, offer a credit card to, but their labeled data may consist of a much smaller set of customers. While the potential of using unlabeled data exists, much research is needed to understand the specific circumstances when it should be used, and the specific methods for exploiting unlabeled data that should be used.

CONCLUSION

There are situations with a large amount of unlabeled data and a small amount of labeled data. Using only labeled data to build classification models can potentially ignore useful information contained in the unlabeled data. How to leverage the information contained in the unlabeled data to help improve the accuracy of the classification model is an important research question. Prior research on learning with unlabeled data focuses either on selecting unlabeled data to acquire labels, or use models built on labeled data to assign labels to unlabeled data. After discussing various approaches in handling unlabeled data for more accurate classification, this chapter focuses on one approach which augments the features of the labeled data to capture information in the unlabeled data in order to achieve higher classification accuracy. The problem of exploiting unlabeled data is relatively new, and is one that can have a wide variety of applications. Further research is needed to understand unaddressed issues for leveraging unlabeled data for more accurate classification.

REFERENCES

Agrawal, R., Mannila, H., Srikant, R., Toivonen, H. & Verkamo, A. I. (1995). Fast Discovery of Association Rules. In *Advances in Knowledge Discovery and Data Mining*, AAAI Press.

Atkinson, A. (1996). The usefulness of optimum experimental designs. *Journal of the Royal Statistical Society, 58(1),* 59-76.

Balcan, M. & Blum, A. (2005). A PAC-Style Model for Learning from Labeled and Unlabeled Data. In *Proceedings of the 18th Annual Conference on Learning Theory*.

Blum, A. & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the 1998 Conference on Computational Learning Theory*.

Chapelle, O., Weston, J. & Schölkopf, B. (2003). Cluster Kernels for Semi-Supervised Learning. *Advances in Neural Information Processing Systems 15*, 585-592.

Dempster, A. P., Laird, N. M. & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of Royal Statistical Society B, 39*, 1-38.

Hakkani-Tur, D., Tur, G., Rahim, M. & Riccardi, G. (2004). Unsupervised and active learning in automatic speech recognition for call classification. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal*.

Han, J., & Kamber, M. (Ed.). (2001). *Data mining: concepts and techniques (The Morgan Kaufmann series in data management systems), 1st Edition.* San Francisco, CA: Morgan Kaufmann Publishers.

Fawcett, T. (1993). *Feature discovery for inductive concept learning*. (Tech. Rep. No. 90-15), University of Massachusetts.

Joachims, T. (1999). Transductive Inference for Text Classification using Support Vector Machines. In *Proceedings of the International Conference on Machine Learning*.

MacKay, D. J. C. (1992). Information-based objective functions for active data selection. *Neural Computation, 4(4)*, 590-604.

Peng, K., Vucetic, S., Han, B., Xie, X., & Obradovic, Z. (2003). Exploiting Unlabeled Data for Improving Accuracy of Predictive Data Mining. In *Proceedings of ICDM*, 267-274.

Raina, R., Battle, A., Lee, H., Packer, B. & Ng, A. Y. (2007). Self-taught learning: transfer learning from unlabeled data. In *Proceedings of the 24th international conference on Machine learning*.

Saar-Tsechansky, M., & Provost, F. J. (2001). Active sampling for class probability estimation and ranking. *Machine Learning, 54(2)*, 153-178.

Seeger, M. (2001). *Leaning with Labeled and Unlabeled Data*. (Tech. Rep.), Edinburgh University, UK.

Yan, R., Yang, J. & Hauptmann, A. (2003). Automatically labeling video data using multi-class active learning. In *Proceedings of Ninth IEEE International Conference on Computer Vision.*

Yang, Y., & Padmanabhan, B. (2007). On the use of patterns for leveraging unlabeled data for classification. University of California, Davis, Working Paper.

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, 189-196.

KEY TERMS

Active Learning: It's a goal-directed data acquisition method for incomplete data. The usual scenario considered in active learning is that all explanatory variables are known, but the value for the target variable is often unknown and expensive to acquire. The problem is to determine which points to acquire this target value from with the specific goal of improving model performance at manageable cost.

Classification: Classification is a form of data analysis that can be used to extract classification models to predict categorical class labels.

Decision Tree: It's a predictive model. In the produced tree structure, leaves represent classifications and branches represent conjunction of features that lead to those classifications.

EM Algorithm: It's used in statistics for finding maximum likelihood estimates of parameters in probabilistic models, where the model depends on unobserved latent variables.

Itemset: A set of items. It's often used in association rule mining. The occurrence frequency of an itemset (a set of items) is the number of transactions that contain the itemset.

Labeled Data: If a data record has a known value for its class label, this data record is called a labeled data.

Unlabeled Data: If the value for a data record's class is unknown, it's called an unlabeled data.

Locally Adaptive Techniques for Pattern Classification

Carlotta Domeniconi
George Mason University, USA

Dimitrios Gunopulos
University of California, USa

INTRODUCTION

Pattern classification is a very general concept with numerous applications ranging from science, engineering, target marketing, medical diagnosis and electronic commerce to weather forecast based on satellite imagery. A typical application of pattern classification is mass mailing for marketing. For example, credit card companies often mail solicitations to consumers. Naturally, they would like to target those consumers who are most likely to respond. Often, demographic information is available for those who have responded previously to such solicitations, and this information may be used in order to target the most likely respondents. Another application is electronic commerce of the new economy. E-commerce provides a rich environment to advance the state-of-the-art in classification because it demands effective means for text classification in order to make rapid product and market recommendations.

Recent developments in data mining have posed new challenges to pattern classification. Data mining is a knowledge discovery process whose aim is to discover unknown relationships and/or patterns from a large set of data, from which it is possible to predict future outcomes. As such, pattern classification becomes one of the key steps in an attempt to uncover the *hidden knowledge* within the data. The primary goal is usually predictive accuracy, with secondary goals being speed, ease of use, and interpretability of the resulting predictive model.

While pattern classification has shown promise in many areas of practical significance, it faces difficult challenges posed by real world problems, of which the most pronounced is Bellman's *curse of dimensionality*: it states the fact that the sample size required to perform accurate prediction on problems with high dimensionality is beyond feasibility. This is because in high dimensional spaces data become extremely sparse and are apart from each other. As a result, severe bias that affects any estimation process can be introduced in a high dimensional feature space with finite samples.

Learning tasks with data represented as a collection of a very large number of features abound. For example, microarrays contain an overwhelming number of genes relative to the number of samples. The Internet is a vast repository of disparate information growing at an exponential rate. Efficient and effective document retrieval and classification systems are required to turn the ocean of bits around us into useful information, and eventually into knowledge. This is a challenging task, since a word level representation of documents easily leads 30000 or more dimensions.

This chapter discusses classification techniques to mitigate the curse of dimensionality and reduce bias, by estimating feature relevance and selecting features accordingly. This issue has both theoretical and practical relevance, since many applications can benefit from improvement in prediction performance.

BACKGROUND

In a classification problem an observation is characterized by q feature measurements $\boldsymbol{x} = \left(x_1, \cdots, x_q\right) \in \Re^q$ and is presumed to be a member of one of J classes, L_j, $j = 1, \cdots, J$. The particular group is unknown, and the goal is to assign the given object to the correct group using its measured features $\boldsymbol{x}$.

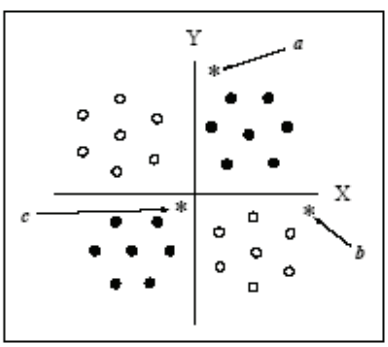
Feature relevance has a *local* nature. Therefore, any chosen fixed metric violates the assumption of locally constant class posterior probabilities, and fails to make correct predictions in different regions of the input space. In order to achieve accurate predictions, it becomes crucial to be able to estimate the different degrees of relevance that input features may have in various locations of the feature space.

Consider, for example, the rule that classifies a new data point with the label of its closest training point in the measurement space (*1-Nearest Neighbor rule*). Suppose each instance is described by 20 features, but only three of them are relevant to classifying a given instance. In this case, two points that have identical values for the three relevant features may nevertheless be distant from one another in the 20-dimensional input space. As a result, the similarity metric that uses all 20 features will be misleading, since the distance between neighbors will be dominated by the large number of irrelevant features. This shows the effect of the curse of dimensionality phenomenon, that is, in high dimensional spaces distances between points within the same class or between different classes may be similar. This fact leads to highly biased estimates. Nearest neighbor approaches (Ho, 1998; Lowe, 1995) are especially sensitive to this problem.

In many practical applications things are often further complicated. In the previous example, the three relevant features for the classification task at hand may be dependent on the location of the query point, i.e. the point to be classified, in the feature space. Some features may be relevant within a specific region, while other features may be more relevant in a different region. Figure 1 illustrates a case in point, where class boundaries are parallel to the coordinate axes. For query *a*, dimension *X* is more relevant, because a slight move along the *X* axis may change the class label, while for query *b*, dimension *Y* is more relevant. For query *c*, however, both dimensions are equally relevant.

These observations have two important implications. Distance computation does not vary with equal strength or in the same proportion in all directions in the feature space emanating from the input query. Moreover, the value of such strength for a specific feature may vary from location to location in the feature space. Capturing such information, therefore, is of great importance to any classification procedure in high dimensional settings.

Figure 1. Feature relevance varies with query locations



MAIN THRUST

Severe bias can be introduced in pattern classification in a high dimensional input feature space with finite samples. In the following we introduce adaptive metric techniques for distance computation, capable of reducing the bias of the estimation.

Friedman (Friedman, 1994) describes an adaptive approach (the Machete and Scythe algorithms) for classification that combines some of the best features of kNN learning and recursive partitioning. The resulting hybrid method inherits the flexibility of recursive partitioning to adapt the shape of the neighborhood $N(\mathbf{x}_0)$ of query $(\mathbf{x}_0)$, as well as the ability of nearest neighbor techniques to keep the points within $N(\mathbf{x}_0)$ close to the point being predicted. The method is capable of producing nearly continuous probability estimates with the region $N(\mathbf{x}_0)$ centered at $(\mathbf{x}_0)$, and the shape of the region separately customized for each individual prediction point.

The major limitation concerning the Machete/Scythe method is that, like recursive partitioning methods, it applies a "greedy" strategy. Since each split is conditioned on its "ancestor" split, minor changes in an early split, due to any variability in parameter estimates, can have a significant impact on later splits, thereby producing different terminal regions. This makes the predictions highly sensitive to the sampling fluctuations associated with the random nature of the process that produces the training data, and therefore may lead to high variance predictions.

In (Hastie & Tibshirani, 1996), the authors propose a discriminant adaptive nearest neighbor classification method (DANN) based on linear discriminant analysis. Earlier related proposals appear in (Myles & Hand, 1990; Short & Fukunaga, 1981). The method in (Hastie & Tibshirani, 1996) computes a local distance metric as a product of weighted within and between sum of squares matrices. The authors also describe a method to perform global dimensionality reduction, by pooling the local dimension information over all points in

the training set (Hastie & Tibshirani, 1996; Hastie & Tibshirani, 1996a).

While sound in theory, DANN may be limited in practice. The main concern is that in high dimensions one may never have sufficient data to fill in $q \times q$ (within and between sum of squares) matrices (where q is the dimensionality of the problem). Also, the fact that the distance metric computed by DANN approximates the weighted *Chi-squared* distance only when class densities are Gaussian and have the same covariance matrix may cause a performance degradation in situations where data do not follow Gaussian distributions or are corrupted by noise, which is often the case in practice.

A different adaptive nearest neighbor classification method (ADAMENN) has been introduced to try to minimize bias in high dimensions (Domeniconi, Peng, & Gunopulos, 2002), and to overcome the above limitations. ADAMENN performs a *Chi-squared* distance analysis to compute a flexible metric for producing neighborhoods that are highly adaptive to query locations. Let $\mathbf{x}$ be the nearest neighbor of a query $\mathbf{x}_0$ computed according to a distance metric $D(\mathbf{x},\mathbf{x}_0)$. The goal is to find a metric $D(\mathbf{x},\mathbf{x}_0)$ that minimizes $E[r(\mathbf{x},\mathbf{x}_0)]$, where $r(\mathbf{x}_0,\mathbf{x}) = \sum_{j=1}^{J} \Pr(j \mid \mathbf{x}_0)(1-\Pr(j \mid \mathbf{x}))$. Here $\Pr(j|\mathbf{x})$ is the class conditional probability at $\mathbf{x}$. That is, $r(\mathbf{x}_0,\mathbf{x})$ is the finite sample error risk given that the nearest neighbor to $\mathbf{x}_0$ by the chosen metric is $\mathbf{x}$. It can be shown (Domeniconi, Peng, & Gunopulos, 2002) that the weighted *Chi-squared* distance

$$D(\mathbf{x},\mathbf{x}_0) = \sum_{j=1}^{J} \frac{[\Pr(j \mid \mathbf{x}) - \Pr(j \mid \mathbf{x}_0)]^2}{\Pr(j \mid \mathbf{x}_0)}. \quad (1)$$

approximates the desired metric, thus providing the foundation upon which the ADAMENN algorithm computes a measure of local feature relevance, as shown below.

The first observation is that $\Pr(j|\mathbf{x})$ is a function of $\mathbf{x}$. Therefore, one can compute the conditional expectation of $\Pr(j|\mathbf{x})$, denoted by $\overline{\Pr}(j \mid x_i = z)$, given that x_i assumes value z, where x_i represents the ith component of $\mathbf{x}$. That is, $\overline{\Pr}(j \mid x_i = z) = E[\Pr(j \mid \mathbf{x}) \mid x_i = z] = \int \Pr(j \mid \mathbf{x})\, p(\mathbf{x} \mid x_i = z) d\mathbf{x}$. Here $p(\mathbf{x}|x_i = z)$ is the conditional density of the other input variables defined as $p(\mathbf{x} \mid x_i = z) = p(\mathbf{x})\delta(x_i - z) / \int p(\mathbf{x})\delta(x_i - z) d\mathbf{x}$, where $\delta(x-z)$ is the Dirac delta function having the properties $\delta(x-z) = 0$ *if* $x \neq z$ and $\int_{-\infty}^{\infty} \delta(x-z)dx = 1$. Let

$$r_i(\mathbf{z}) = \sum_{j=1}^{J} \frac{[\Pr(j \mid \mathbf{z}) - \overline{\Pr}(j \mid x_i = z_i)]^2}{\overline{\Pr}(j \mid x_i = z_i)}. \quad (2)$$

$r_i(\mathbf{z})$ represents the ability of feature i to predict the $\Pr(j|\mathbf{z})$ s at $x_i = z_i$. The closer $\overline{\Pr}(j \mid x_i = z_i)$ is to $\Pr(j|\mathbf{z})$, the more information feature i carries for predicting the class posterior probabilities locally at $\mathbf{z}$.

We can now define a measure of feature relevance for $\mathbf{x}_0$ as

$$\bar{r}_i(\mathbf{x}_0) = \frac{1}{K} \sum_{\mathbf{z} \in N(\mathbf{x}_0)} r_i(\mathbf{z}), \quad (3)$$

where $N(\mathbf{x}_0)$ denotes the neighborhood of $\mathbf{x}_0$ containing the K nearest training points, according to a given metric. $\bar{r}_i$ measures how well on average the class posterior probabilities can be approximated along input feature i within a local neighborhood of $\mathbf{x}_0$. Small $\bar{r}_i$ implies that the class posterior probabilities will be well approximated along dimension i in the vicinity of $\mathbf{x}_0$. Note that $\bar{r}_i(\mathbf{x}_0)$ is a function of both the test point $\mathbf{x}_0$ and the dimension i, thereby making $\bar{r}_i(\mathbf{x}_0)$ a local relevance measure in dimension i.

The relative relevance, as a weighting scheme, can then be given by $w_i(\mathbf{x}_0) = \frac{R_i(\mathbf{x}_0)^t}{\sum_{l=1}^{q} R_l(\mathbf{x}_0)^t}$, where $t = 1,2$, giving rise to linear and quadratic weightings respectively, and $R_i(\mathbf{x}_0) = \max_{j=1}^{q}\{\bar{r}_j(\mathbf{x}_0)\} - \bar{r}_i(\mathbf{x}_0)$ (i.e., the larger the R_i, the more relevant dimension i). We propose the following exponential weighting scheme

$$w_i(\mathbf{x}_0) = \exp(cR_i(\mathbf{x}_0)) / \sum_{l=1}^{q} \exp(cR_l(\mathbf{x}_0)) \quad (4)$$

where c is a parameter that can be chosen to maximize (minimize) the influence of $\bar{r}_i$ on w_i. When $c = 0$ we have $w_i = 1/q$, which has the effect of ignoring any difference among the $\bar{r}_i$'s. On the other hand, when c is large a change in $\bar{r}_i$ will be exponentially reflected in w_i. The exponential weighting is more sensitive to changes in local feature relevance and in general gives rise to better performance improvement. In fact, it is more stable because it prevents neighborhoods from extending infinitely in any direction, i.e., zero weight. This, however, can occur when either linear or quadratic weighting is used. Thus, equation (4) can be used to compute the weight associated with each feature, resulting in the weighted distance computation:

$$D(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{i=1}^{q} w_i (x_i - y_i)^2}. \quad (5)$$

The weights $\mathbf{w}_i$ enable the neighborhood to elongate less important feature dimensions, and, at the same time, to constrict the most influential ones. Note that the technique is *query-based* because the weights depend on the query (Aha, 1997; Atkeson, Moore, & Shaal, 1997).

An intuitive explanation for (2) and, hence, (3), goes as follows. Suppose that the value of $r_i(\mathbf{z})$ is small, which implies a large weight along dimension *i*. Consequently, the neighborhood is shrunk along that direction. This, in turn, penalizes points along dimension *i* that are moving away from z_i. Now, $r_i(\mathbf{z})$ can be small only if the subspace spanned by the other input dimensions at $x_i = z_i$ likely contains samples similar to $\mathbf{z}$ in terms of the class conditional probabilities. Then, a large weight assigned to dimension *i* based on (4) says that moving away from the subspace, hence from the data similar to $\mathbf{z}$, is not a good thing to do. Similarly, a large value of $r_i(\mathbf{z})$, hence a small weight, indicates that in the vicinity of z_i along dimension *i* one is unlikely to find samples similar to $\mathbf{z}$. This corresponds to an elongation of the neighborhood along dimension *i*. Therefore, in this situation in order to better predict the query, one must look farther away from z_i.

One of the key differences between the relevance measure (3) and Friedman's is the first term in the squared difference. While the class conditional probability is used in (3), its expectation is used in Friedman's. This difference is driven by two different objectives: in the case of Friedman's, the goal is to seek a dimension along which the expected variation of $\Pr(j|\mathbf{x})$ is maximized, whereas in (3) a dimension is found that minimizes the difference between the class probability distribution for a given query and its conditional expectation along that dimension (2). Another fundamental difference is that the machete/scythe methods, like recursive partitioning, employ a greedy peeling strategy that removes a subset of data points permanently from further consideration. As a result, changes in an early split, due to any variability in parameter estimates, can have a significant impact on later splits, thereby producing different terminal regions. This makes predictions highly sensitive to the sampling fluctuations associated with the random nature of the process that produces the training data, thus leading to high variance predictions. In contrast, ADAMENN employs a "patient" averaging strategy that takes into account not only the test point $\mathbf{x}_0$ itself, but also its K_0 nearest neighbors. As such, the resulting relevance estimates (3) are in general more robust and have the potential to reduce the variance of the estimates.

In (Hastie & Tibshirani, 1996), the authors show that the resulting metric approximates the weighted *Chi-squared* distance (1) by a Taylor series expansion, given that class densities are Gaussian and have the same covariance matrix. In contrast, ADAMENN does not make such assumptions, which are unlikely in real world applications. Instead, it attempts to approximate the weighted *Chi-Squared* distance (1) directly. The main concern with DANN is that in high dimensions we may never have sufficient data to fill in $q \times q$ matrices. It is interesting to note that the ADAMENN algorithm can potentially serve as a general framework upon which to develop a unified adaptive metric theory that encompasses both Friedman's work and that of Hastie and Tibshirani.

FUTURE TRENDS

Almost all problems of practical interest are high dimensional. With the recent technological trends, we can expect an intensification of research effort in the area of feature relevance estimation and selection. In bioinformatics, the analysis of microarray data poses challenging problems. Here one has to face the problem of dealing with more dimensions (genes) than data points (samples). Biologists want to find "marker genes" that are differentially expressed in a particular set of conditions. Thus, methods that simultaneously cluster genes and samples are required to find distinctive "checkerboard" patterns in matrices of gene expression data. In cancer data, these checkerboards correspond to genes that are up- or down-regulated in patients with particular types of tumors. Increased research effort in this area is needed and expected.

Clustering is not exempted from the curse of dimensionality. Several clusters may exist in different subspaces, comprised of different combinations of features. Since each dimension could be relevant to at least one of the clusters, global dimensionality reduction techniques are not effective. We envision further investigation on this problem with the objective of developing robust techniques in presence of noise.

Recent developments on kernel-based methods suggest a framework to make the locally adaptive techniques discussed above more general. One can perform feature relevance estimation in an induced

feature space, and then use the resulting kernel metrics to compute distances in the input space. The key observation is that kernel metrics may be non-linear in the input space, but are still linear in the induced feature space. Hence, the use of suitable non-linear features allows the computation of locally adaptive neighborhoods with arbitrary orientations and shapes in input space. Thus, more powerful classification techniques can be generated.

CONCLUSIONS

Pattern classification faces a difficult challenge in finite settings and high dimensional spaces due to the curse of dimensionality. In this chapter we have presented and compared techniques to address data exploration tasks such as classification and clustering. All methods design adaptive metrics or parameter estimates that are local in input space in order to dodge the curse of dimensionality phenomenon. Such techniques have been demonstrated to be effective for the achievement of accurate predictions.

REFERENCES

Aha, D. (1997). Lazy Learning. *Artificial Intelligence Review*, 11:1-5.

Atkeson, C., Moore, A. W., & Schaal, S. (1997). Locally Weighted Learning. *Artificial Intelligence Review*, 11:11-73.

Domeniconi, C., Peng, J., & Gunopulos, D. (2002). Locally adaptive metric nearest neighbor classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(9):1281-1285.

Friedman, J. H. (1994). Flexible metric nearest neighbor classification. *Technical Report*, Department of Statistics, Stanford University.

Hastie, T., & Tibshirani, R (1996). Discriminant adaptive nearest neighbor classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **18**(6):607-615.

Hastie, T., & Tibshirani, R. (1996). Discriminant Analysis by Gaussian Mixtures. *Journal of the Royal Statistical Society*, 58:155-176.

Ho, T. K. (1998). Nearest Neighbors in Random Subspaces. *Joint IAPR International Workshops on Advances in Pattern Recognition*.

Lowe, D. G. (1995). Similarity Metric Learning for a Variable-Kernel Classifier. *Neural Computation*, 7(1):72-85.

Myles, J. P. & Hand, D. J. (1990). The Multi-Class Metric Problem in Nearest Neighbor Discrimination Rules. *Pattern Recognition*, 23(11):1291-1297.

Short, R. D. & Fukunaga, K. (1981). Optimal Distance Measure for Nearest Neighbor Classification. *IEEE Transactions on Information Theory*, 27(5):622-627.

KEY TERMS

Classification: The task of inferring concepts from observations. It is a mapping from a measurement space into the space of possible meanings, viewed as finite and discrete target points (class labels). It makes use of training data.

Clustering: The process of grouping objects into subsets, such that those within each cluster are more closely related to one another than objects assigned to different clusters, according to a given similarity measure.

Curse of Dimensionality: Phenomenon that refers to the fact that in high dimensional spaces data become extremely sparse and are far apart from each other. As a result, the sample size required to perform an accurate prediction in problems with high dimensionality is usually beyond feasibility.

Kernel Methods: Pattern analysis techniques that work by embedding the data into a high dimensional vector space, and by detecting linear relations in that space. A kernel function takes care of the embedding.

Local Feature Relevance: Amount of information that a feature carries to predict the class posterior probabilities at a given query.

Nearest Neighbor Methods: Simple approach to the classification problem. It finds the *K* nearest neighbors of the query in the training set, and then predicts the class label of the query as the most frequent one occurring in the *K* neighbors.

Pattern: A structure that exhibits some form of regularity, able to serve as a model representing a concept of what was observed.

Recursive Partitioning: Learning paradigm that employs local averaging to estimate the class posterior probabilities for a classification problem.

Subspace Clustering: Simultaneous clustering of both row and column sets in a data matrix.

Training Data: Collection of observations (characterized by feature measurements), each paired with the corresponding class label.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 684-688, copyright 2005 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

Mass Informatics in Differential Proteomics

Xiang Zhang
University of Louisville, USA

Seza Orcun
Purdue University, USA

Mourad Ouzzani
Purdue University, USA

Cheolhwan Oh
Purdue University, USA

INTRODUCTION

Systems biology aims to understand biological systems on a comprehensive scale, such that the components that make up the whole are connected to one another and work in harmony. As a major component of systems biology, differential proteomics studies the differences between distinct but related proteomes such as normal versus diseased cells and diseased versus treated cells. High throughput mass spectrometry (MS) based analytical platforms are widely used in differential proteomics (Domon, 2006; Fenselau, 2007). As a common practice, the proteome is usually digested into peptides first. The peptide mixture is then separated using multidimensional liquid chromatography (MDLC) and is finally subjected to MS for further analysis. Thousands of mass spectra are generated in a single experiment. Discovering the significantly changed proteins from millions of peaks involves mass informatics. This paper introduces data mining steps used in mass informatics, and concludes with a descriptive examination of concepts, trends and challenges in this rapidly expanding field.

BACKGROUND

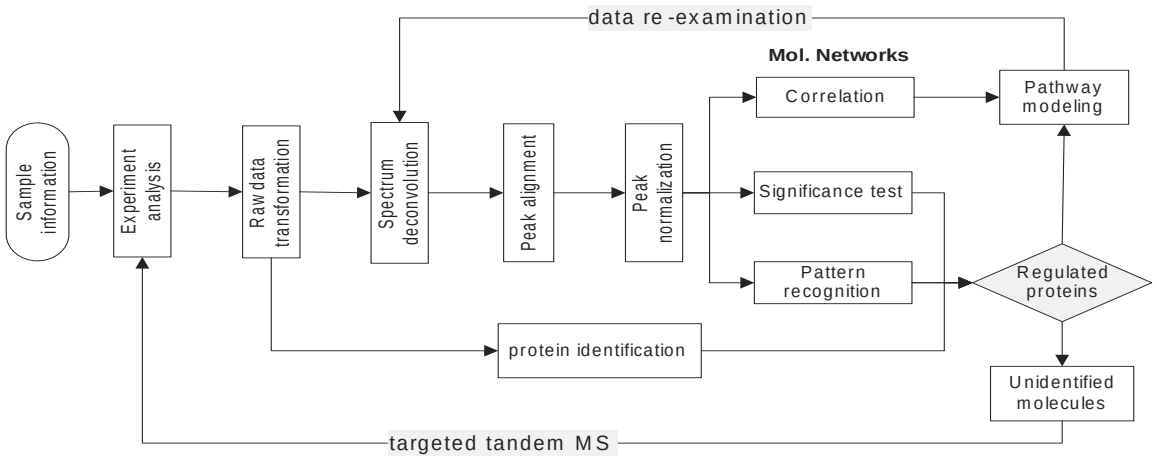
Proteomics was initially envisioned as a technique to globally and simultaneously characterize all components in a proteome. In recent years, a rapidly emerging set of key technologies is making it possible to identify large numbers of proteins in a mixture or complex, to map their interactions in a cellular context, and to analyze their biological activities. Several MS based technologies have emerged for identifying large numbers of proteins expressed in cells and globally detecting the differences in levels of proteins in different cell states (Asara, 2006).

Due to the complexity of the proteome, a major effort in proteomics research is devoted to fractionation of proteins and peptides prior to MS. One way to fractionate peptide mixtures to the extent required for MS analysis of the component peptides is to couple multiple chromatography columns in tandem (Hattan, 2005; Qui, 2007). The fractionated peptide mixture is then subjected to MS for further separation and mass analysis. Thousands of mass spectra will be generated in a single differential proteomics experiment. Mass informatics is involved in identifying the significantly changed proteins from millions of peptide peaks.

DATA MINING FRAMEWORK

A variety of mass spectrometers are commercially available. Each of these mass spectrometers stores raw mass spectra in a proprietary format. The raw spectra have to be transformed into common data format first. As in any study of biological phenomena, it is crucial that only relevant observations are identified and related to each other. The interpretation and comprehension of the collection of mass spectra presents major challenges and involve several data mining steps. The aim of mass informatics is to reduce data dimensionality and to extract relevant knowledge from thousands of mass spectra (Arneberg, 2007). Figure 1 shows an overall framework for mass informatics in differential proteomics. Most of the components of this framework

Figure 1. Information flow chart in differential proteomics



will be discussed in this paper with the exception of the pathway modeling.

Spectra Deconvolution

The purpose of spectra deconvolution is to differentiate signals arising from the real analyte as opposed to signals arising from contaminants or instrumental noise, and to reduce data dimensionality which will benefit down stream statistical analysis. Therefore, spectra deconvolution extracts peak information from thousands of raw mass spectra. The peak information is reported in a simple peak table. As an example, GISTool (Zhang, 2005a) is a software package using chemical noise filtering, charge state fitting, and de-isotoping for the analysis of complex peptide samples. Overlapping peptide signals in mass spectra are deconvoluted by correlating the observed spectrum with modeled peptide isotopic peak profiles. Isotopic peak profiles for peptides were generated *in silico* from a protein database producing reference model distributions. Several other spectra deconvolution packages such as RelEx (MacCoss, 2003), MSQuant (http://msquant.sourceforge.net/), and ASAPRatio (Li, 2003) have been developed to find quantitative information about proteins and peptides.

Protein Identification

Two methods are currently used for protein identification: database searching and *de novo* sequencing. Database searching correlates the spectra with protein sequences. The database-searching algorithm starts with spectrum reduction to remove chemical noise. A list of peptides is generated *in silico* from the protein database using enzyme specificity. After applying potential chemical modifications, *in silico* peptides that have similar molecular weight to the precursor ions of tandem mass spectrum (MS/MS) are selected as candidate peptides. A theoretical spectrum is then created for each candidate peptide and these theoretical spectra are compared with the experimental spectrum. A final ranking list is generated using different scoring functions. Disadvantages of the database searching strategy are very well understood: the protein of interest might not be present in the sequence database, prediction errors are present in gene-finding programs, the protein database may not be available in some cases, genes might undergo alternative splicing resulting in novel proteins, and amino acids may mutate and undergo unknown modifications. SEQUEST (Eng, 1994) and MASCOT (Perkins, 1999) are two database searching software packages used most frequently.

de novo sequencing derives the peptide sequence directly from the tandem mass spectrum (Kanazawa, 2007). Lutefisk (Taylor, 1997) is a popular *de novo*

sequencing software. The Lutefisk algorithm translates the spectrum into a "spectrum graph" where the nodes in the graph correspond to peaks in the spectrum; two nodes are connected by an edge if the mass difference between the two corresponding peaks is equal to the mass of an amino acid. The software then attempts to find a path that connects the N- and C-termini, and to connect all the nodes corresponding to the y-ions or b-ions. After all the sequences are obtained, a scoring procedure is used to rank the sequences. The advantage of *de novo* sequencing is that it does not rely on a protein database. However, it requires high mass accuracy. Furthermore, it cannot correctly assign peptides because of amino acid mass equivalence.

Peak Alignment

Ideally, the same peptide detected on the same analytical system should have the same value of the measurement. For example, if a peptide is measured on a liquid chromatography mass spectrometry (LC-MS) system, retention time and molecular weight of this peptide, in different samples, should be the same. However, this may not be the case due to experimental variation. The objective of peak alignment is to recognize peaks of the same molecule occurring in different samples from millions of peaks detected during the course of an experiment. XAlign software (Zhang, 2005b) uses a two-step alignment approach. The first step addresses systematic retention time shift by recognizing and aligning significant peaks. A significant peak refers to a peak that is present in every sample and is the most intense peak in a certain m/z and retention time range. Discrete convolution is used in the second step to align overlapped peaks. The other alignment software packages include LCMSWARP (Jaitly, 2006) and PETAL (Wang, 2007).

Normalization

To allow multi-experiment analyses, it is important to first normalize the data to make the samples comparable. Normalization step targets to quantitatively filter overall peak intensity variations due to experimental errors such as varying amounts of samples loaded onto LC-MS. Several normalization methods have been proposed. One is to choose an analysis run as reference and individually normalize all others relative to this reference, one at a time (Wang, 2003). The intensity ratio of each aligned peak pair of a given analysis run and the reference is calculated. The normalization constant for the analysis run being considered is then taken as the median of the ratios of intensities for all components between the analysis run in question and the reference analysis run. Zhu et al. normalized the MS data by dividing the intensity at each m/z value by the average intensity of the entire spectrum (Zhu, 2003). The log linear model method (Hartemink, 2001) assumes primarily multiplicative variation. The maximum likelihood and maximum *a posteriori* estimates for the parameters characterizing the multiplicative variation were derived to compute the scaling factors needed for normalization.

Statistical Significance Test

The purpose of statistical significance test is to identify peptide peaks that make significant contributions to the protein profile of a sample, or that distinguish a group of samples from others. The procedure can be summarized into the following criteria for biomarker discovery:

Qualitative analysis for significant peaks that are present in one group but not in the other. Qualitative difference indicates the situation in which a peak is present only in few samples, e.g. less than 50% of the samples, in one group but is present in most of the samples in the other group. In this situation, a binary variable is used to denote the presence of this peak in each sample. The comparison between two groups is then performed through a contingency table, whose columns and rows correspond to groups versus presence/absence, respectively. A chi-square test provides an adequate test statistic about whether the presence of this peak is significantly different between the two groups or not.

Statistical tests for quantitatively different peaks between two groups. In addition to qualitative differences, some peaks may be present in multiple sample groups but their intensity might differ between the groups. The quantitative difference indicates the situation in which a peak is present in most (or all) of the samples, but has different intensities between the groups. In this situation, the standard two-sample t-test or the Wilcoxon-Mann-Whitney rank test can be used to compare the group differences.

Pattern Recognition

Many types of pattern recognition systems fall into two main categories, supervised and unsupervised. Supervised systems require knowledge or data in which the outcome or classification is known ahead of time, so that the system can be trained to recognize and distinguish outcomes. Unsupervised systems cluster or group records without previous knowledge of outcome or classification.

The most frequently used approach of unsupervised pattern recognition is principal component analysis (PCA). PCA's main function is to reduce dimensions of the multivariate, multi-channeled data to a manageable subset, a new set of uncorrelated variables called principle components (PCs). These PCs serves as an approximation to the original data and allows an analyst to overview the data in the reduced dimension space and study the major constituents of the overall variability in the data. Other unsupervised methods include hierarchical clustering, k-means, and self organizing maps (SOM).

Many algorithms perform supervised learning, such as discriminant function analysis, partial least square, artificial neural networks, nearest-neighbor. The most popular supervised learning system used in differential proteomics is support vector machine (SVM). The SVM is fundamentally a binary classifier. It operates by mapping the given training set into a possible high-dimensional feature space and attempting to locate in that space a hyperplane that separates the positive examples from the negative ones. Having found such a plane, the SVM can then predict the classification of an unlabeled example by mapping it into the feature space and determining on which side of the separating plane the example lies. Much of the SVM's power comes from its criterion for selecting a separating hyperplane when candidate planes exist: the SVM chooses the plane that maintains a maximum margin from any point in the training set.

Protein Correlation Networks

Protein correlation studies the relationships between protein concentrations. For example, two proteins will have a negative correlation if the concentration of one protein increases while the other decreases in the same sample. Protein correlation not only reveal important relationships among the various proteins, but also provides information about the biochemical processes underlying the disease or drug response. Correlations of biological molecules may be linear or non-linear in nature; a common evaluation approach is to estimate molecular correlations by calculating the Pearson's correlation coefficient.

Interactive visualization of protein correlation networks is one of the major informatic components in differential proteomic data analysis. Many software packages have been developed for interactive visualization such as Cytoscape (Shannon, 2003), Graphviz (http://www.graphviz.org/), CFinder (Adamcsek, 2006), and Tom Sawyer (http://www.tomsawyer.com). These programs can be used to display protein correlation networks. However, these software packages are limited in providing data mining capability. SysNet (Zhang, 2007) is an interactive visual data mining application for 'omics expression data analysis, which is able to integrate molecular expression data obtained from different 'omics experiments, interactively analyze intermolecular correlations using different statistical models, and perform interactive analysis of time lapse data to assess molecular evolution.

FUTURE TRENDS AND CHALLENGES

The study of differential proteomics has much to offer to scientists with high confidence. Intelligent data mining facilities, however, still need to be developed to prevent important results from being lost in the massive information. There is much less work done in bridging the wet laboratory research and data mining. This kind of research includes experimental design, quality assurance, preliminary data analysis, etc. Differential proteomics also requires informatics tools to accurately and rapidly query databases to select relevant results. Biological databases are characterized by large quantities of data exhibiting different characteristics, e.g. data models, attributes grouping, and semantics. The technology to organize, search, integrate, and evolve these sources has not kept pace with the rapid growth of the available information space. Scientists usually go from one database to another and manually query them and consolidate the results of their queries. The efficient sharing of data is especially challenging in such environments where the information sources are largely autonomous, heterogeneous, fragmented and evolve dynamically.

Differential proteomics also faces challenge in high confidence protein identification. Typically ~10-20% of tandem spectra from LC-MS/MS analyses can be confidently identified while the remaining spectra are typically ignored or discarded. Several factors contribute to this inefficiency: imperfect scoring schema, criteria that constrain the protein database search, incorrect mass and charge assignment, and low-quality MS/MS spectra. To date, the physical properties of peptides exploited for peptide selection, or fractionation are not widely considered or utilized for protein identification even though these methods are routinely used in proteomics. Another problem confounding current proteomics efforts is false positive protein identifications. In the recent years, it became apparent that peptide and protein assignments obtained from protein database searches using tandem MS data were subject to the assignment of many false positives within a dataset.

Data visualization is another challenge in bioinformatics. Results in differential proteomics are extremely complicated even after data mining. Presenting findings in a way that biologists can easily understand and extract the information by themselves is crucial for the success of differential proteomics. A good example of this is bio-networks. In a single differential proteomics experiment, bio-informatician can build a bio-network that contains thousands of correlated proteins and related chemical components. It will be a nightmare for a biologist to extract useful information from this massive network unless an intelligent informatic tool is provided.

CONCLUSION

High throughput mass spectrometry based analytical platforms generate massive amount of data for differential proteomics research. Several data mining steps are involved in extracting scientific information from such data. These data mining steps include spectrum deconvolution, protein identification, peak alignment, normalization, statistical significance test, pattern recognition, and protein correlation networks. Each of these data mining steps faces a variety of challenges toward efficient and accurate data analysis. Many ongoing efforts to overcome these generic inefficiencies have achieved varying degrees of success. However, challenges in this rapidly expanding field of mass informatics are still preventing scientists from fully understanding the proteome.

REFERENCES

Adamcsek, B., Palla, G., Farkas, I. J., Derényi, I., & Vicsek, T. (2006). CFinder: Locating cliques and overlapping modules in biological networks. *Bioinformatics*, 22, 1021-1023.

Arneberg, R., Rajalahti, T., Flikka, K., Berven, F. S., Kroksveen, A. C., Berle, M., Myhr, K, Vedeler, C. A., Ulvik, R. J. & Kvalheim, O. M. (2007). Pretreatment of mass spectral profiles: application to proteomic data. *Anal. Chem.*, 79, 7014-7026.

Asara, J. M., Zhang, X., Zheng, B., Christofk, H. H., Wu, N. & Cantley, L. C. (2006). In-gel stable isotope labeling for relative quantification using mass spectrometry. *Nature Protocols*., 1, 46-51.

Fenselau, C. (2007). A review of quantitative methods for proteomic studies. *J. Chromatogr., B.* 855, 14-20.

Domon, B. & Aebersold, R. (2006). Mass spectrometry and protein analysis, Science, 312, 212-217.

Eng, J. K., McCormack, A. L. & Yates, J. R. III. (1994). An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database, *J. Am. Soc. Mass Spectrom.*, 5, 976-989.

Geng, M., Zhang, X., Bina, M. & Regnier, F. E. (2001). Proteomics of glycoproteins based on affinity selection of glycopeptides from tryptic digests. *J. Chromatogr. B.*, 752, 293-306.

Hartemink, A. J., Gifford, D. K., Jaakola, T. S. & Young, R. A. (2001). Maximum likelihood estimation of optimal scaling factors for expression array normalization, *Proceedings of SPIE,* 4266.

Hattan, S. J., Marchese, J., Khainovski, N., Martin, S., Juhasz, P. (2005). Comparative study of [three] LC-MALDI workflows for the analysis of complex proteomic samples, *J. Proteome Res., 4,* 1931-1941.

Jaitly, N., Monroe, M. E., Petyuk, V. A., Clauss, T. R. W., Adkins, J. N. & Smith, R. D. (2006). Robust algorithm for alignment of liquid chromatography-mass spectrometry analyses in an accurate mass and time tag data analysis pipeline, *Anal. Chem.*, 78, 7397-7409.

Mitsuhiro Kanazawa, M., Anyoji, H., Ogiwara, A. & Nagashima. U. (2007). De novo peptide sequencing using ion peak intensity and amino acid cleavage intensity ratio, *Bioinformatics*, 23, 1063-1072.

MacCoss, M. J., Wu, C. C., Liu, H., Sadygov, R. & Yates, J. R. III. (2003). A correlation algorithm for the automated quantitative analysis of shotgun proteomics data, *Anal. Chem*., 75, 6912-6921.

Li, X., Zhang, H., Ranish, J. A. & Aebersold, R. (2003). Automated statistical analysis of protein abundance ratios from data generated by stable-isotope dilution and tandem mass spectrometry, *Anal. Chem*., 75, 6648-6657.

Wang, P., Tang, H., Fitzgibbon, M. P., McIntosh, M., Coram, M., Zhang, H., Yi, E., Aebersold, R. (2007). A statistical method for chromatographic alignment of LC-MS data. *Biostatistics*, 8, 357-367.

Perkins, D. N., Pappin, D. J. C., Creasy, D. M., Cottrell, J. S. (1999). Probability-based protein identification by searching sequence databases using mass spectrometry data, *Electrophoresis*, 20, 35551-3567.

Qiu, R., Zhang, X. & Regnier, F. E. (2007). A method for the identification of glycoproteins from human serum by a combination of lectin affinity chromatography along with anion exchange and Cu-IMAC selection of tryptic peptides. *J. Chromatogr. B.*, 845, 143-150.

Shannon, P., Markiel, A., Ozier, O., Baliga, N. S., Wang, J. T., Ramage, D., Amin, N., Schwikowski, B. & Ideker, T. (2003). Cytoscape: a software environment for integrated models of biomolecular interaction networks, *Genome Res.*, 13, 2498-2504.

Taylor, J. A. & Johson, R. S. (1997). Sequence database searches via de novo peptide sequencing by tandem mass spectrometry, *Rapid Commun. Mass Spectrum.*, 11, 1067-1075.

Wang, W., Zhou, H., Lin, H., Roy, S., Shaler, T. A., Hill, L. R., Norton, S., Kumar, P., Anderle, M. & Becker, C. H. (2003). Quantification of proteins and metabolites by mass spectrometry without isotopic labeling or spiked standards, *Anal. Chem.* 75, 4818-4826.

Zhang, X., Hines, W., Adamec, J., Asara, J., Naylor, S. & Regnier, F. E. (2005a). An automated method for the analysis of stable isotope labeling data for proteomics, *J. Am. Soc. Mass Spectrom.*, 16, 1181-1191.

Zhang, X., Asara, J. M., Adamec, J., Ouzzani, M. & Elmagarmid, A. K. (2005b). Data preprocessing in liquid chromatography mass spectrometry based proteomics, *Bioinformatics*, 21, 4054-4059.

Zhang, M., Ouyang, Q., Stephenson, A., Kane, M. D., Salt, D. E., Prabhakar, S., Buck, C. & Zhang, X. (2007). Interactive analysis of 'omics molecular expression data, *BMC Systems Biology*, submitted.

Zhu, W., Wang, X., Ma, Y., Rao, M., Glimm, J. & Kovach J. S. (2003). Detection of cancer-specific markers amid massive mass spectral data, *PNAS*, 100, 14666-14671.

KEY TERMS

Differential Proteomics: A research field that qualitatively and quantitatively compares proteomes under different conditions to further unravel biological processes.

Interactive Data Mining: A data mining process that discovers scientific knowledge through interactive communication between human and computer.

Liquid Chromatography: An analytical chromatographic technique that is useful for separating ions or molecules that are dissolved in a solvent.

Mass Informatics: Bioinformatics research that focuses on data mining of experimental data generated from mass spectrometry.

Mass Spectrometry: An analytical technique used to measure the mass-to-charge ratio of ions.

Protein Biomarkers: Naturally occurring proteins that can be used for measuring the prognosis and/or progress of diseases and therapies.

Proteome: The entire complement of proteins expressed by a genome, cell, tissue or organism at a particular time and under specific conditions.

Systems Biology: A field in biology aiming at system level understanding of biological systems by studying the interactions between their different components. It attempts to create predictive models of cells, organs, biochemical processes and complete organisms.

Materialized View Selection for Data Warehouse Design

Dimitri Theodoratos
New Jersey Institute of Technology, USA

Wugang Xu
New Jersey Institute of Technology, USA

Alkis Simitsis
National Technical University of Athens, Greece

INTRODUCTION

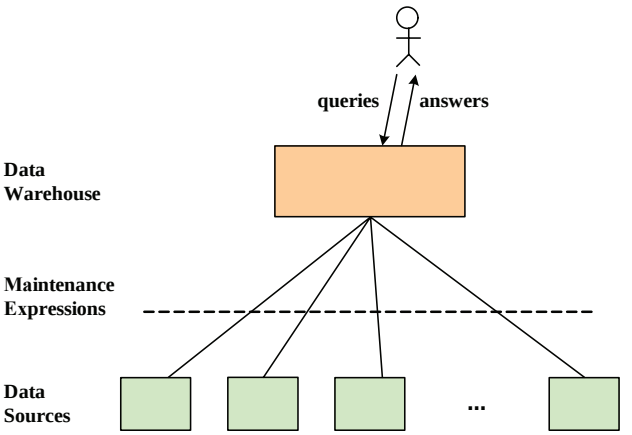
A Data Warehouse (DW) is a repository of information retrieved from multiple, possibly heterogeneous, autonomous, distributed databases and other information sources for the purpose of complex querying, analysis and decision support. Data in the DW are selectively collected from the sources, processed in order to resolve inconsistencies, and integrated in advance (at design time) before data loading. DW data are usually organized multidimensionally to support On-Line Analytical Processing (OLAP). A DW can be abstractly seen as a set of materialized views defined over the source relations. During the initial design of a DW, the DW designer faces the problem of deciding which views to materialize in the DW. This problem has been addressed in the literature for different classes of queries and views and with different design goals.

BACKGROUND

Figure 1 shows a simplified DW architecture. The DW contains a set of materialized views. The users address their queries to the DW. The materialized views are used partially or completely for evaluating the user queries. This is achieved through partial or complete rewritings of the queries using the materialized views.

When the source relations change, the materialized views need to be updated. The materialized views are usually maintained using an incremental strategy. With such a strategy, the changes to the source relations are propagated to the DW. The changes to the materialized views are computed using the changes of the source relations, and are eventually applied to the materialized views. The expressions used to compute the view changes involve the changes of the source relations,

Figure 1. A simplified DW architecture



and are called maintenance expressions. Maintenance expressions are issued by the DW against the data sources and the answers are sent back to the DW. When the source relation changes affect more than one materialized view, multiple maintenance expressions need to be evaluated. The techniques of multiquery optimization can be used to detect ``common subexpressions'' among maintenance expressions in order to derive an efficient global evaluation plan for all the maintenance expressions.

MAIN THRUST OF THE CHAPTER

When selecting views to materialize in a DW, one attempts to satisfy one or more design goals. A design goal is either the minimization of a cost function or a constraint. A constraint can be classified as user oriented or system oriented. Attempting to satisfy the constraints can result in no feasible solution to the view selection problem. The design goals determine the design of the algorithms that select views to materialize from the space of alternative view sets.

Minimization of Cost Functions

Most approaches comprise in their design goals the minimization of a cost function. The following are examples of cost functions to be minimized:

- **Query evaluation cost.** Often, the queries that the DW has to satisfy are given as input to the view selection problem. The overall query evaluation cost is the sum of the cost of evaluating each input query rewritten (partially or completely) over the materialized views. This sum can also be weighted, each weight indicating the frequency or importance of the corresponding query. Several approaches aim at minimizing the query evaluation cost (Harinarayan et al., 1996; Shukla et al 1998; Gupta & Mumick, 1999).
- **View maintenance cost.** The view maintenance cost is the sum of the cost of propagating each source relation change to the materialized views. This sum can be weighted, each weight indicating the frequency of propagation of the changes of the corresponding source relation. The maintenance expressions can be evaluated more efficiently if they can be partially rewritten over views already materialized at the DW: the evaluation of parts of the maintenance expression is avoided since their materializations are present at the DW. Moreover, access of the remote data sources and expensive data transmissions are reduced. Materialized views that are added to the DW for reducing the view maintenance cost are called *auxiliary views* (Ross et al., 1996; Theodoratos & Sellis, 1999). The auxiliary views can be materialized permanently or transiently. Transiently materialized views are used during the maintenance process and discarded afterwards (Mistry et al., 2001). Obviously, maintaining the auxiliary views incurs additional maintenance cost. However, if this cost is less than the reduction to the maintenance cost of the initially materialized views, it is worth keeping the auxiliary views in the DW. Ross et al. (1996) derive auxiliary views to permanently materialize in order to minimize the view maintenance cost.
- **Operational cost.** Minimizing the query evaluation cost and the view maintenance cost are conflicting requirements. Low view maintenance cost can be obtained by replicating source relations at the DW. In this case, though, the query evaluation cost is high since queries need to be computed from the replicas of the source relations. Low query evaluation cost can be obtained by materializing at the DW all the input queries. In this case, all the input queries can be answered by a simple lookup but the view maintenance cost is high since complex maintenance expressions over the source relations need to be computed. The input queries may overlap, that is, they may share many common subexpressions. By materializing common subexpressions and other views over the source relations, it is possible, in general, to reduce the view maintenance cost. These savings must be balanced against higher query evaluation cost. For this reason, one can choose to minimize a linear combination of the query evaluation and view maintenance cost which is called *operational cost*. Most approaches endeavor to minimize the operational cost (Gupta, 1997; Baralis et al., 1997; Yang et al., 1997; Theodoratos & Sellis, 1999).
- **Total view size.** In a distributed DW where the materialized views are stored remotely, the performance bottleneck is usually the data transmission time over the network. In this case, the designer

is interested to minimize the size of the set of materialized views that answer all input queries (Chirkova et al., 2006).

System Oriented Constraints

System oriented constraints are dictated by the restrictions of the system, and are transparent to the users.

- **Space constraint.** Although the degradation of the cost of disk space allows for massive storage of data, one cannot consider that the disk space is unlimited. The space constraint restricts the space occupied by the selected materialized views not to exceed the space allocated to the DW for this end. Space constraints are adopted in many works (Harinarayan et al; 1996, Gupta, 1997; Theodoratos & Sellis, 1999; Golfarelli & Rizzi, 2000).
- **View maintenance cost constraint.** In many practical cases the refraining factor in materializing all the views in the DW is not the space constraint but the view maintenance cost. Usually, DWs are updated periodically, e.g. at nighttime, in a large batch update transaction. Therefore the update window must be sufficiently short so that the DW is available for querying and analysis during daytime. The view maintenance cost constraint states that the total view maintenance cost should be less than a given amount of view maintenance time. Gupta & Mumick, (1999), Golfareli & Rizzi (2000), and Lee & Hammer (2001) consider a view maintenance cost constraint in selecting materialized views.
- **Self Maintainability.** A materialized view is self-maintainable if it can be maintained, for any instance of the source relations over which it is defined, and for all source relation changes, using only these changes, the view definition, and the view materialization. The notion is extended to a set of views in a straightforward manner. By adding auxiliary views to a set of materialized views, one can make the whole view set self-maintainable. There are different reasons for making a view set self-maintainable: (a) The remote source relations need not be contacted in turn for evaluating maintenance expressions during view updating. (b)''Anomalies'' due to concurrent changes are eliminated and the view maintenance process is simplified. (c) The materialized views can be maintained efficiently even if the sources are not able to answer queries (e.g. legacy systems), or if they are temporarily unavailable (e.g. in mobile systems). By adding auxiliary views to a set of materialized views, the whole view set can be made self-maintainable. Self-maintainability can be trivially achieved by replicating at the DW all the source relations used in the view definitions. Self-maintainability viewed as a constraint requires that the set of materialized views taken together is self-maintainable. Quass et al (1996), Akinde et al (1998), Liang et al (1999) and Theodoratos (2000) aim at making the DW self-maintainable.
- **Answering the input queries using exclusively the materialized views.** This constraint requires the existence of a complete rewriting of the input queries (which are initially defined over the source relations) over the materialized views. Clearly, if this constraint is satisfied, the remote data sources need not be contacted for evaluating queries. This way, expensive data transmissions from the DW to the sources and conversely are avoided. Some approaches assume a centralized DW environment where the source relations are present at the DW site. In this case the answerability of the queries from the materialized views is trivially guaranteed by the presence of the source relations. The answerability of the queries can also be trivially guaranteed by appropriately defining select-project views on the source relations, and replicating them at the DW. This approach guarantees at the same time the self-maintainability of the materialized views. Theodoratos & Sellis (1999) do not assume a centralized DW environment or replication of part of the source relations at the DW, and explicitly impose this constraint in selecting views for materialization.

User Oriented Constraints

User oriented constraints express requirements of the users.

- **Answer data currency constraints.** An answer data currency constraint sets an upper bound on the time elapsed between the point in time the answer to a query is returned to the user and the

point in time the most recent changes of a source relation that are taken into account in the computation of this answer are read (this time reflects the currency of answer data). Currency constraints are associated with every source relation in the definition of every input query. The upper bound in an answer data currency constraint (minimal currency required) is set by the users according to their needs. This formalization of data currency constraints allows stating currency constraints at the query level and not at the materialized view level as is the case in some approaches. Therefore, currency constraints can be exploited by DW view selection algorithms where the queries are the input, while the materialized views are the output (and therefore are not available). Furthermore, it allows stating different currency constraints for different relations in the same query.

- **Query response time constraints.** A query response time constraint states that the time needed to evaluate an input query using the views materialized at the DW should not exceed a given bound. The bound for each query is given by the users and reflects their needs for fast answers. For some queries fast answers may be required, while for others the response time may not be predominant.

Search Space and Algorithms

Solving the problem of selecting views for materialization involves addressing two main tasks: (a) generating a search space of alternative view sets for materialization, and (b) designing optimization algorithms that select an optimal or near-optimal view set from the search space.

A DW is usually organized according to a star or snow-flake schema where a fact table is surrounded by a number of dimension tables. The dimension tables define hierarchies of aggregation levels. Typical OLAP queries involve star joins (key/foreign key joins between the fact table and the dimension tables) and grouping and aggregation at different levels of granularity. For queries of this type, the search space can be formed in an elegant way as a multidimensional lattice (Harinarayan et al., 1996; Baralis et al., 1997).

Gupta (1997) states that the view selection problem is NP-hard. Most of the approaches on view selection problems avoid exhaustive algorithms. The adopted algorithms fall in two categories: deterministic and randomized. In the first category belong greedy algorithms with performance guarantee (Harinarayan et al., 1996; Gupta, 1997), 0-1 integer programming algorithms (Yang et al., 1997), A* algorithms (Gupta & Mumick, 1999), and various other heuristic algorithms (Ross et al., 1996; Baralis et al., 1997; Shukla et al., 1998; Theodoratos & Sellis, 1999). In the second category belong simulated annealing algorithms (Theodoratos et al., 2001; Kalnis et al., 2002), iterative improvement algorithms (Kalnis et al., 2002) and genetic algorithms (Lee & Hammer, 2001). Both categories of algorithms exploit the particularities of the specific view selection problem and the restrictions of the class of queries considered.

FUTURE TRENDS

The view selection problem has been addressed for different types of queries. Research has focused mainly on queries over star schemas. Newer applications, e.g. XML or web based applications, require different types of queries. This topic has only been partially investigated (Labrinidis & Roussopoulos, 2000; Golfarelli et al., 2001).

A relevant issue that needs further investigation is the construction of the search space of alternative view sets for materialization. Even though the construction of such a search space for grouping and aggregation queries is straightforward (Harinarayan et al., 1966), it becomes an intricate problem for general queries (Golfarelli & Rizzi, 2001).

Indexes can be seen as special types of views. Gupta et al (1997) show that a two-step process that divides the space available for materialization and picks views first and then indexes can perform very poorly. More work needs to be done on the problem of automating the selection of views and indexes together.

DWs are dynamic entities that evolve continuously over time. As time passes, new queries need to be satisfied. A dynamic version of the view selection problem chooses additional views for materialization, and avoids the design of the DW from scratch (Theodoratos & Sellis, 2000). A system that dynamically materializes views in the DW at multiple levels of granularity in order to match the workload (Kotidis &

Roussopoulos, 2001; Lawrence, M. & Rau-Chaplin, 2006) is a current trend in the design of a DW. In some applications the workload comprises query and update statements that need to be executed in a certain order (that is, the workload is a sequence of statements). In this case, a view management scheme can be employed that exploits the statement sequence information, and dynamically creates and drops transiently materialized views. Such a scheme can achieve further reductions to the operational cost of the DW (Agrawal et al., 2006; Xu et al., 2007).

CONCLUSION

A DW can be seen as a set of materialized views. A central problem in the design of a DW is the selection of views to materialize in it. Depending on the requirements of the prospective users of the DW, the materialized view selection problem can be formulated with various design goals which comprise the minimization of cost functions and the satisfaction of user and system oriented constraints. Because of its importance, different versions of it have been the focus of attention of many researchers in recent years. Papers in the literature deal mainly with the issue of determining a search space of alternative view sets for materialization and with the issue of designing optimization algorithms that avoid examining exhaustively the usually huge search space. Some results of this research have already been used in commercial database management systems (Agrawal et al., 2004; Zilio et al. 2004).

REFERENCES

Agrawal , S., Chaudhuri, S., Kollar, L., Marathe, A. P., Narasayya, V. R., & Syamala, M. (2004). Database Tuning Advisor for Microsoft SQL Server. *International Conference on Very Large Data Bases* (VLDB), Toronto, Canada, 1110–1121.

Agrawal S., Chu E. & Narasayya Narasayya V. R. (2006). Automatic physical design tuning: workload as a sequence. *ACM SIGMOD International Conference on Management of Data (SIGMOD).* Chicago, Illinois, 683-694.

Akinde, M. O., Jensen, O. G., & Böhlen, H. M. (1998). Minimizing Detail Data in Data Warehouses. *International Conference on Extending Database Technology (EDBT)*. Valencia, Spain, 293-307.

Baralis, E., Paraboschi, S., & Teniente, E. (1997). Materialized Views Selection in a Multidimensional Database. *International Conference on Very Large Data Bases*. Athens, Greece, 156-165.

Golfarelli, M., & Rizzi, S. (2000). View materialization for nested GPSJ queries. *International Workshop on Design and Management of Data Warehouses (DMDW)*. Stockholm, Sweden, 1-10.

Golfarelli, M., Rizzi, S., & Vrdoljak B. (2001, November). Data Warehouse Design from XML Sources. *ACM International Workshop on Data Warehousing and OLAP (DOLAP*). Atlanta, USA.

Gupta, H. (1997). Selection of Views to Materialize in a Data Warehouse. *International Conference on Database Theory (ICDT)*. Delphi, Greece, 98-112.

Gupta, H., Harinarayan, V., Rajaraman, A., & Ullman, J. D. (1997, April). Index Selection for OLAP. *IEEE International Conference on Data Engineering*, Birmingham, UK, 208-219.

Gupta, H., & Mumick, I.S. (1999). Selection of Views to Materialize Under a Maintenance Cost Constraint. *International Conference on Database Theory (ICDT)*. Jerusalem, Israel, 453-470.

Harinarayan, V., Rajaraman, A., & Ullman, J. (1996). Implementing Data Cubes Efficiently. *ACM SIGMOD International Conference on Management of Data (SIGMOD).* Montreal, Canada, 205-216.

Kalnis, P., Mamoulis, N., & Papadias, D. (2002). View selection using randomized search. *Data & Knowledge Engineering*, 42(1), 89-111.

Kotidis, Y., & Roussopoulos, N. (2001). A case for dynamic view management. *ACM Transactions on Database Systems*, 26(4), 388-423.

Lawrence, M. & Rau-Chaplin, M. (2006). Dynamic View Selection for OLAP. *International conference on Data Warehousing and Knowledge Discovery (DaWaK).* Krakow, Poland, 33-44.

Labrinidis, A., & Roussopoulos, N. (2000). WebView Materialization. *ACM SIGMOD International Conference on Management of Data (SIGMOD)*. Dallas,

USA, 367-378

Lee, M., & Hammer, J. (2001). Speeding Up Materialized View Selection in Data Warehouses Using a Randomized Algorithm. *International Journal of Cooperative Information Systems (IJCIS)*,10(3), 327-353.

Liang, W. (1999). Making Multiple Views Self-Maintainable in a Data Warehouse. *Data & Knowledge Engineering*, 30(2), 121-134.

Mistry H., Roy P., Sudarshan S. & Ramamritham K. (2001). Materialized View Selection and Maintenance Using Multi-Query Optimization. *ACM SIGMOD International Conference on Management of Data (SIGMOD).* Santa Barbara, California, 307-318.

Quass, D., Gupta, A., Mumick, I.S., & Widom, J. (1996). Making Views Self-Maintainable for Data Warehousing. *International Conference on Parallel and Distributed Information Systems (PDIS)*. Florida, USA,158-169.

Chirkova, R. & Li, C. & Li, J. (2006). Answering queries using materialized views with minimum size. *The International Journal on Very Large Data Bases (VLDB J)*. 191-210.

Ross, K., Srivastava, D., & Sudarshan, S. (1996). Materialized View Maintenance and Integrity Constraint Checking: Trading Space for Time. *ACM SIGMOD International Conference on Management of Data (SIGMOD).* Montreal, Canada, 447-458.

Shukla, A., Deshpande, P., & Naughton, J. (1998). Materialized View Selection for Multidimensional Datasets. *International Conference on Very Large Data Bases (VLDB)*. New York City, USA, 488-499.

Theodoratos, D. (2000). Complex View Selection for Data Warehouse Self-Maintainability. *International Conference on Cooperative Information Systems (CoopIS)*. Eilat, Israel, 78-89.

Theodoratos, D., Dalamagas, T., Simitsis, A., & Stavropoulos, M. (2001). A Randomized Approach for the Incremental Design of an Evolving Data Warehouse. *International Conference on Conceptual Modeling (ER)*. Yokohama, Japan, 325-338.

Theodoratos, D., & Sellis, T. (1999). Designing Data Warehouses. *Data & Knowledge Engineering*, 31(3), 279-301.

Theodoratos, D., & Sellis, T. (2000). Incremental Design of a Data Warehouse. *Journal of Intelligent Information Systems (JIIS)*, 15(1), 7-27.

Xu, W. & Theodoratos D. & Zuzarte, C. & Wu, X. & Oria V. (2007). A Dynamic View Materialization Scheme for Sequences of Queries. *International conference on Data Warehousing and Knowledge Discovery (DaWaK)*. Regensberg Gemany, 55-65.

Yang, J., Karlapalem, K., & Li, Q. (1997). Algorithms for Materialized View Design in Data Warehousing Environment. *International Conference on Very Large Data Bases*. Athens, Greece, 136-145.

Zilio, D. C., et al. (2004). Recommending Materialized Views and Indexes with IBM DB2 Design Advisor. *International Conference on Autonomic Computing (ICAC)*. New York, NY, USA, 180–188.

KEY TERMS

Auxiliary View: A view materialized in the DW exclusively for reducing the view maintenance cost. An auxiliary view can be materialized permanently or transiently.

Materialized View: A view whose answer is stored in the DW.

Operational Cost: A linear combination of the query evaluation and view maintenance cost.

Query Evaluation Cost: The sum of the cost of evaluating each input query rewritten over the materialized views.

Self-Maintainable View: A materialized view that can be maintained, for any instance of the source relations, and for all source relation changes, using only these changes, the view definition, and the view materialization.

View: A named query.

View Maintenance Cost: The sum of the cost of propagating each source relation change to the materialized views.

Matrix Decomposition Techniques for Data Privacy

Jun Zhang
University of Kentucky, USA

Jie Wang
University of Kentucky, USA

Shuting Xu
Virginia State University, USA

INTRODUCTION

Data mining technologies have now been used in commercial, industrial, and governmental businesses, for various purposes, ranging from increasing profitability to enhancing national security. The widespread applications of data mining technologies have raised concerns about trade secrecy of corporations and privacy of innocent people contained in the datasets collected and used for the data mining purpose. It is necessary that data mining technologies designed for knowledge discovery across corporations and for security purpose towards general population have sufficient privacy awareness to protect the corporate trade secrecy and individual private information. Unfortunately, most standard data mining algorithms are not very efficient in terms of privacy protection, as they were originally developed mainly for commercial applications, in which different organizations collect and own their private databases, and mine their private databases for specific commercial purposes.

In the cases of inter-corporation and security data mining applications, data mining algorithms may be applied to datasets containing sensitive or private information. Data warehouse owners and government agencies may potentially have access to many databases collected from different sources and may extract any information from these databases. This potentially unlimited access to data and information raises the fear of possible abuse and promotes the call for privacy protection and due process of law.

Privacy-preserving data mining techniques have been developed to address these concerns (Fung et al., 2007; Zhang, & Zhang, 2007). The general goal of the privacy-preserving data mining techniques is defined as to hide sensitive individual data values from the outside world or from unauthorized persons, and simultaneously preserve the underlying data patterns and semantics so that a valid and efficient decision model based on the distorted data can be constructed. In the best scenarios, this new decision model should be equivalent to or even better than the model using the original data from the viewpoint of decision accuracy. There are currently at least two broad classes of approaches to achieving this goal. The first class of approaches attempts to distort the original data values so that the data miners (analysts) have no means (or greatly reduced ability) to derive the original values of the data. The second is to modify the data mining algorithms so that they allow data mining operations on distributed datasets without knowing the exact values of the data or without direct accessing the original datasets. This article only discusses the first class of approaches. Interested readers may consult (Clifton et al., 2003) and the references therein for discussions on distributed data mining approaches.

BACKGROUND

The input to a data mining algorithm in many cases can be represented by a vector-space model, where a collection of records or objects is encoded as an $n \times m$ object-attribute matrix (Frankes, & Baeza-Yates, 1992). For example, the set of vocabulary (words or terms) in a dictionary can be the items forming the rows of the matrix, and the occurrence frequencies of all terms in a document are listed in a column of the matrix. A

collection of documents thus forms a term-document matrix commonly used in information retrieval. In the context of privacy-preserving data mining, each column of the data matrix can contain the attributes of a person, such as the person's name, income, social security number, address, telephone number, medical records, etc. Datasets of interest often lead to a very high dimensional matrix representation (Achlioptas, 2004). It is observable that many real-world datasets have nonnegative values for attributes. In fact, many of the existing data distortion methods inevitably fall into the context of matrix computation. For instance, having the longest history in privacy protection area and by adding random noise to the data, additive noise method can be viewed as a random matrix and therefore its properties can be understood by studying the properties of random matrices (Kargupta et al., 1991).

Matrix decomposition in numerical linear algebra typically serves the purpose of finding a computationally convenient means to obtain the solution to a linear system. In the context of data mining, the main purpose of matrix decomposition is to obtain some form of simplified low-rank approximation to the original dataset for understanding the structure of the data, particularly the relationship within the objects and within the attributes and how the objects relate to the attributes (Hubert, Meulman, & Heiser, 2000). The study of matrix decomposition techniques in data mining, particularly in text mining, is not new, but the application of these techniques as data distortion methods in privacy-preserving data mining is a recent interest (Xu et al., 2005). A unique characteristic of the matrix decomposition techniques, a compact representation with reduced-rank while preserving dominant data patterns, stimulates researchers' interest in utilizing them to achieve a win-win goal both on high degree privacy-preserving and high level data mining accuracy.

MAIN FOCUS

Data distortion is one of the most important parts in many privacy-preserving data mining tasks. The desired distortion methods must preserve data privacy, and at the same time, must keep the utility of the data after the distortion (Verykios et al., 2004). The classical data distortion methods are based on the random value perturbation (Agrawal, & Srikant, 2000). The more recent ones are based on the data matrix-decomposition strategies (Wang et al., 2006; Wang et al., 2007; Xu et al., 2006).

Uniformly Distributed Noise

The original data matrix A is added with a uniformly distributed noise matrix E_u. Here E_u is of the same dimension as that of A, and its elements are random numbers generated from a continuous uniform distribution on the interval from C_1 to C_2. The distorted data matrix A_u is denoted as: $A_u = A + E_u$.

Normally Distributed Noise

Similar to the previous method, here the original data matrix A is added with a normally distributed noise matrix E_n, which has the same dimension as that of A. The elements of E_n are random numbers generated from the normal distribution with a parameter mean μ and a standard deviation ρ. The distorted data matrix A_n is denoted as: $A_n = A + E_n$.

Singular Value Decomposition

Singular Value Decomposition (SVD) is a popular matrix factorization method in data mining and information retrieval. It has been used to reduce the dimensionality of (and remove the noise in the noisy) datasets in practice (Berry et al., 1999). The use of SVD technique in data distortion is proposed in (Xu et al., 2005). In (Wang et al., 2007), the SVD technique is used to distort portions of the datasets.

The SVD of the data matrix A is written as:

$$A = U\Sigma V^T$$

where U is an $n \times n$ orthonormal matrix, $\Sigma = \text{diag}[\sigma_1, \sigma_2, \cdots, \sigma_s]$ ($s = \min\{m,n\}$) is an $n \times m$ diagonal matrix whose nonnegative diagonal entries (the singular values) are in a descending order, and V^T is an $m \times m$ orthonormal matrix. The number of nonzero diagonal entries of Σ is equal to the rank of the matrix A.

Due to the arrangement of the singular values in the matrix Σ (in a descending order), the SVD transformation has the property that the maximum variation among the objects is captured in the first dimension, as $\sigma_1 \geq \sigma_i$ for $i \geq 2$. Similarly, much of the remaining variation is

captured in the second dimension, and so on. Thus, a transformed matrix with a much lower dimension can be constructed to represent the structure of the original matrix faithfully. Define:

$$A_k = U_k \Sigma_k V_k^T$$

where U_k contains the first k columns of U, Σ_k contains the first k nonzero singular values, and V_k^T contains the first k rows of V^T. The rank of the matrix A_k is k. With k being usually small, the dimensionality of the dataset has been reduced dramatically from $\min\{m,n\}$ to k (assuming all attributes are linearly independent). It has been proved that A_k is the best k dimensional approximation of A in the sense of the Frobenius norm.

In data mining applications, the use of A_k to represent A has another important implication. The removed part $E_k = A - A_k$ can be considered as the noise in the original dataset (Xu et al., 2006). Thus, in many situations, mining on the reduced dataset A_k may yield better results than mining on the original dataset A. When used for privacy-preserving purpose, the distorted dataset A_k can provide protection for data privacy, at the same time, it keeps the utility of the original data as it can faithfully represent the original data structure.

Sparsified Singular Value Decomposition

After reducing the rank of the SVD matrices, we can further distort the data matrices by removing their small size entries. This can be done with a threshold strategy. Given a threshold value ε, we set any data entry in the matrices U_k and V_k^T to be zero if its absolute value is smaller than ε. We refer to this operation as the dropping operation (Gao, & Zhang, 2003). For example, we set $u_{ij} = 0$ in U_k if $|u_{ij}| < \varepsilon$. Similar operation is applied to the entries in V_k^T. Let $\overline{U_k}$ denote U_k with dropped entries and $\overline{V_k^T}$ denote V_k^T with dropped entries, we can represent the distorted dataset as:

$$\overline{A_k} = \overline{U_k} \Sigma_k \overline{V_k^T}$$

The sparsified SVD method is equivalent to further distorting the dataset A_k. Denoting $E_\varepsilon = A_k - \overline{A_k}$, we have:

$$A = \overline{A_k} + E_k + E_\varepsilon$$

The dataset provided to the data mining analysts is $\overline{A_k}$, which is twice distorted in the sparsified SVD strategy. Without the knowledge of E_k and E_ε, it will be difficult for the data mining analysts to recover the exact values of A, based on the disclosed values of A_k.

Nonnegative Matrix Factorization

Given an $n \times m$ nonnegative matrix dataset A with $A_{ij} \geq 0$ and a prespecified positive integer $k \leq \min\{n,m\}$, the Nonnegative Matrix Factorization (NMF) finds two nonnegative matrices $W \in R^{n \times k}$ with $W_{ij} \geq 0$ and $H \in R^{k \times m}$ with $H_{ij} \geq 0$, such that $A \approx WH$ and the objective function:

$$f(W,H) = \frac{1}{2} \| A - WH \|_F^2$$

is minimized. Here $\|\bullet\|_F$ is the Frobenius norm. The matrices W and H may have many other desirable properties in data mining applications. Several algorithms to compute nonnegative matrix factorizations for some applications of practical interests are proposed in (Lee, & Seung, 1999; Pascual-Montano et al., 2006). Some of these algorithms are modified in (Wang et al., 2006) to compute nonnegative matrix factorizations for enabling privacy-preserving in datasets for data mining applications. Similar to the sparsified SVD techniques, sparsification techniques can be used to drop small size entries from the computed matrix factors to further distort the data values (Wang et al., 2006).

In text mining, NMF has an advantage over SVD in the sense that if the data values are nonnegative in the original dataset, NMF maintains their nonnegativity, but SVD does not. The nonnegativity constraints can lead to a parts-based representation because they allow only additive, not subtractive, combinations of the original basis vectors (Lee, & Seung, 1999). Thus, dataset values from NMF have some meaningful interpretations in the original sense. On the contrary, data values from SVD are no longer guaranteed to be nonnegative. There has been no obvious meaning for the negative values in the SVD matrices. In the context of privacy-preserving, on the other hand, the negative values in the dataset may actually be an advantage, as they further obscure the properties of the original datasets.

Utility of the Distorted Data

Experiments with both SVD and NMF data distortion techniques have been conducted on both synthetic and real-world datasets with a support vector machine classification algorithm. These datasets include some terrorist database, the well-known Wisconsin breast cancer database, and a random number matrix. In these experiments, only the numerical attribute values were distorted. Experimental results reported in (Wang et al., 2007; Wang et al., 2006; Xu et al., 2006; Xu et al., 2005) show that both SVD and NMF techniques provide much higher degree of data distortion than the standard data distortion techniques based on adding uniformly distributed noise or normally distributed noise. In terms of the accuracy of the data mining algorithm, techniques based on adding uniformly distributed noise or normally distributed noise sometimes degrade the accuracy of the classification results, compared with applying the algorithm on the original, undistorted datasets. On the other hand, both SVD and NMF techniques can generate distorted datasets that are able to yield better classification results, compared with applying the algorithm directly on the original, undistorted datasets. This is amazing, as people would intuitively expect that data mining algorithms applied on the distorted datasets may produce less accurate results, than applied on the original datasets.

It is not very clear why the distorted data from SVD and NMF are better for the data classification algorithm used to obtain the experimental results. The hypothesis is that both SVD and NMF may have some functionalities to remove the noise from the original datasets by removing small size matrix entries. Thus, the distorted datasets from SVD and NMF look like "cleaned" datasets. The distorted datasets from the techniques based on adding either uniformly distributed noise or normally distributed noise do not have this property. They actually generate "noisy" datasets in order to distort data values.

FUTURE TRENDS

Using matrix decomposition-based techniques in data distortion for privacy-preserving data mining is a relatively new trend. This class of data privacy-preserving approaches has many desirable advantages over the more standard privacy-preserving data mining approaches. There are a lot of unanswered questions in this new research direction. For example, a classical problem in SVD-based dimensionality reduction techniques is to determine the optimal rank of the reduced dataset matrix. Although in the data distortion applications, the rank of the reduced matrix does not seem to sensitively affect the degree of the data distortion or the level of the accuracy of the data mining results (Wang et al., 2007), it is still of both practical and theoretical interests to be able to choose a good rank size for the reduced data matrix.

Unlike the data distortion techniques based on adding either uniformly distributed noise or normally distributed noise, SVD and NMF does not maintain some statistical properties of the original sestets, such as the mean of the data attributes. Such statistical properties may or may not be important in certain data mining applications. It would be desirable to design some matrix decomposition-based data distortion techniques that maintain these statistical properties.

The SVD and NMF data distortion techniques have been used with the support vector machine based classification algorithms (Xu et al., 2006). It is not clear if they are equally applicable to other data mining algorithms. It is certainly of interest for the research community to experiment these data distortion techniques with other data mining algorithms.

There is also a need to develop certain techniques to quantify the level of data privacy preserved in the data distortion process. Although some measures for data distortion and data utility are defined in (Xu et al., 2006), they are not directly related to the concept of privacy-preserving in datasets.

CONCLUSION

We have presented two classes of matrix decomposition-based techniques for data distortion to achieve privacy-preserving in data mining applications. These techniques are based on matrix factorization techniques commonly practiced in matrix computation and numerical linear algebra. Although their application in text mining is not new, their application in data distortion with privacy-preserving data mining is a recent attempt. Previous experimental results have demonstrated that these data distortion techniques are highly effective for high accuracy privacy protection, in the sense that they can provide high degree of data distortion and

maintain high level data utility with respect to the data mining algorithms.

The computational methods for SVD and NMF are well developed in the matrix computation community. Very efficient software packages are available either in standard matrix computation packages such as MATLAB or from several websites maintained by individual researchers. The availability of these software packages greatly accelerates the application of these and other matrix decomposition and factorization techniques in data mining and other application areas.

REFERENCES

Achlioptas, D. (2004). Random matrices in data analysis. *Proceedings of the 15th European Conference on Machine Learning*, pp. 1-8, Pisa, Italy.

Agrawal, R., & Srikant, R. (2000). Privacy-preserving data mining. *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, pp. 439-450, Dallas, TX.

Berry, M. W., Drmac, Z., & Jessup, E. R. (1999). Matrix, vector space, and information retrieval. *SIAM Review*, 41, 335-362.

Clifton, C., Kantarcioglu, M., Vaidya, J., Lin, X., & Zhu, M. (2003). Tools for privacy preserving distributed data mining. *ACM SIGKDD Explorations*, 4(2), 1-7.

Frankes, W., & Baeza-Yates, R. (1992). *Information Retrieval: Data Structures and Algorithms*. Prentice-Hall, Englewood Cliffs, NJ.

Fung, B.C.M., Wang, K., & Yu, P.S. (2007). Anonymizing classification data for privacy preservation. *IEEE Transactions on Knowledge and Data Engineering*, 19(5), 711-725.

Gao, J., & Zhang, J. (2003). Sparsification strategies in latent semantic indexing. *Proceedings of the 2003 Text Mining Workshop*, pp. 93-103, San Francisco, CA.

Hubert, L., Meulman, J., & Heiser, W. (2000). Two purposes for matrix factorization: a historical appraisal. *SIAM Review*, 42(4), 68-82.

Kargupta, H., Sivakumar, K., & Ghosh, S. (2002). Dependency detection in mobimine and random matrices. *Proceedings of the 6th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pp. 250-262, Helsinki, Finland.

Lee, D. D., & Seung, H. S. (1999). Learning in parts of objects by non-negative matrix factorization. *Nature*, 401, 788-791.

Mahta, M. L. (1991). *Random Matrices*. 2nd edition. Academic, London.

Pascual-Montano, A., Carazo, J. M., Kochi, K., Lehmann, D., & Pascual-Marqui, P. D. (2006). Nonsmooth nonnegative matrix factorization (nsNMF). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28, 403-415.

Verykios, V.S., Bertino, E., Fovino, I. N., Provenza, L. P., Saygin, Y., & Theodoridis, Y. (2004). State-of-the-art in privacy preserving data mining. *ACM SIGMOD Record*, 3(1), 50-57.

Wang, J., Zhong, W. J., & Zhang, J. (2006). NNMF-based factorization techniques for high-accuracy privacy protection on non-negative-valued datasets. *Proceedings of the IEEE Conference on Data Mining 2006*, International Workshop on Privacy Aspects of Date Mining (PADM 2006), pp. 513-517, Hong Kong, China.

Wang, J., Zhang, J., Zhong, W. J., & Xu, S. (2007). A novel data distortion approach via selective SSVD for privacy protection. *International Journal of Information and Computer Security*, to appear.

Xu, S., Zhang, J., Han, D., & Wang, J. (2006). Singular value decomposition based data distortion strategy for privacy protection. *Knowledge and Information Systems*, 10(3), 383-397.

Xu, S., Zhang, J., Han, D., & Wang, J. (2005). Data distortion for privacy protection in a terrorist analysis system. *Proceedings of the 2005 IEEE International Conference on Intelligence and Security Informatics*, pp. 459-464, Atlanta, GA.

Zhang, N, & Zhao, W. (2007). Privacy-preserving data mining systems. *Computer*, 40(4), 52.

KEY TERMS

Data Distortion: A systematic perturbation of data values in a database in order to mask the original data

values, but allow certain properties of the database to be preserved.

Data Utility: A dataset's ability to maintain its performance with data mining algorithms after the data distortion process.

Matrix Decomposition: A factorization of a matrix into some canonical form, usually in the form of a product of two or more matrices.

Nonnegative Matrix Factorization: A class of algorithms that factor a (usually nonnegative) matrix into the product of two matrices, both have nonnegative entries. This type of factorization of matrices is not unique by incorporating different constraints.

Privacy-Preserving Data Mining: Extracting valid knowledge and information from the datasets without learning the underlying data values or data patterns, or without revealing the values or patterns of the private data.

Singular Value Decomposition: The factorization of a rectangular matrix into the product of three matrices. The first and the third matrices are orthonormal. The second matrix is diagonal and contains the singular values of the original matrix.

Measuring the Interestingness of News Articles

Raymond K. Pon
University of California - Los Angeles, USA

Alfonso F. Cardenas
University of California - Los Angeles, USA

David J. Buttler
Lawrence Livermore National Laboratory, USA

INTRODUCTION

An explosive growth of online news has taken place. Users are inundated with thousands of news articles, only some of which are interesting. A system to filter out uninteresting articles would aid users that need to read and analyze many articles daily, such as financial analysts and government officials.

The most obvious approach for reducing the amount of information overload is to learn keywords of interest for a user (Carreira et al., 2004). Although filtering articles based on keywords removes many irrelevant articles, there are still many uninteresting articles that are highly relevant to keyword searches. A relevant article may not be interesting for various reasons, such as the article's age or if it discusses an event that the user has already read about in other articles.

Although it has been shown that collaborative filtering can aid in personalized recommendation systems (Wang et al., 2006), a large number of users is needed. In a limited user environment, such as a small group of analysts monitoring news events, collaborative filtering would be ineffective.

The definition of what makes an article interesting – or its "interestingness" – varies from user to user and is continually evolving, calling for adaptable user personalization. Furthermore, due to the nature of news, most articles are uninteresting since many are similar or report events outside the scope of an individual's concerns. There has been much work in news recommendation systems, but none have yet addressed the question of what makes an article interesting.

BACKGROUND

Working in a limited user environment, the only available information is the article's content and its metadata, disallowing the use of collaborative filtering for article recommendation. Some systems perform clustering or classification based on the article's content, computing such values as TF-IDF weights for tokens (Radev et al., 2003). Corso (2005) ranks articles and new sources based on several properties, such as mutual reinforcement and freshness, in an online method. However, Corso does not address the problem of personalized news filtering, but rather the identification of interesting articles for the general public. Macskassy and Provost (2001) measure the interestingness of an article as the correlation between the article's content and real-life events that occur after the article's publication. Using these indicators, they can predict future interesting articles. Unfortunately, these indicators are often domain specific and are difficult to collect for the online processing of articles.

The online recommendation of articles is closely related to the adaptive filtering task in TREC (Text Retrieval Conference), which is the online identification of articles that are most relevant to a set of topics. The task is different from identifying interesting articles for a user because an article that is relevant to a topic may not necessarily be interesting. However, relevancy to a set of topics of interest is often correlated to interestingness. The report by Robertson and Soboroff (2002) summarizes the results of the last run of the TREC filtering task. Methods explored in TREC11 include a Rocchio variant, a second-order perceptron, a SVM, a Winnow classifier, language modelling, probabilistic models of terms and relevancy, and the Okapi Basic Search System.

The recommendation of articles is a complex document classification problem. However, most classification methods have been used to bin documents into topics, which is a different problem from binning documents by their interestingness. Traditional classification has focused on whether or not an article is relevant to a topic of interest, such as the work done in TREC. Typical methods have included the Rocchio (1971) algorithm, language models (Peng et al., 2003), and latent Dirichlet allocation (Newman et al., 2006; Steyvers, 2006). Despite the research done in topic relevancy classification, it is insufficient for addressing the problem of interestingness. There are many reasons why an article is interesting besides being relevant to topics of interests. For example, an article that discusses content that a user has never seen may be interesting but would be undetectable using traditional IR techniques. For example, the events of the September 11 attacks had never been seen before but were clearly interesting. Furthermore, redundant yet relevant articles would not be interesting as they do not provide the user any new information. However, traditional IR techniques are still useful as a first step towards identifying interesting articles.

MAIN FOCUS

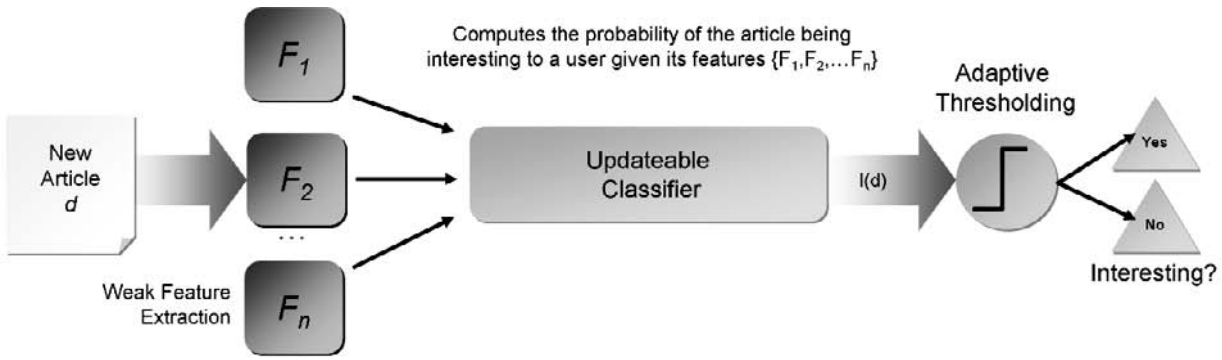
The problem of recommending articles to a specific user can be addressed by answering what makes an article interesting to the user. A possible classification pipeline is envisioned in Figure 1. Articles are processed in a streaming fashion, like the document processing done in the adaptive filter task in TREC. Articles are introduced to the system in chronological order of their publication date. The article classification pipeline consists of four phases. In the first phase, a set of feature extractors generate a set of feature scores for an article. Each feature extractor addresses an aspect of interestingness, such as topic relevancy. Then a classifier generates an overall classification score, which is then thresholded by an adaptive thresholder to generate a binary classification, indicating the interestingness of the article to the user. In the final phase, the user examines the article and provides his own binary classification of interestingness (i.e., label). This feedback is used to update the feature extractors, the classifier, and the thresholder. The process continues similarly for the next document in the pipeline.

Interestingness Issues

The "interestingness" of an article varies from user to user and is often complex and difficult to measure. Consequently, several issues arise:

1. There are a variety of reasons why an article is interesting. There is no single attribute of a document that definitively identifies interesting articles. As a result, using only traditional IR techniques for document classification is not sufficient (Pon et al, 2007).
2. Some interestingness features are often contradictory. For example, an interesting article should be relevant to a user's known interests but should yield new information. On the other hand, random events may be new and unique but may not necessarily be of interest to all users.

Figure 1. Article classification pipeline



3. The breaking news of an important event is difficult to discriminate from the breaking news of an unimportant one.
4. Because what makes an article interesting varies from user to user, the ideal set of features for a user can not be determined until the system is in use by a user. Useless features will be present in the classification process, which will degrade the performance of a classifier (Forman, 2004), especially its accuracy with classifying on early articles.
5. The definition of the interestingness may change for a user over time. Consequently, an online learner must be able to adapt to the changing utility of features.
6. User-feedback must be continually incorporated in the classification process so that any machine learning algorithm can learn efficiently over time what makes an article interesting for a user. A classifier must be incrementally accurate, updateable, and robust against noisy and potentially useless features.
7. Users are often interested in a multitude of topics that may be drastically different from one another. For example, a user may be interested in news about an election and football. To represent a user using a single profile may not be sufficient while multiple profiles may be costly to maintain (Pon et al., 2007b).
8. A successful news recommendation system must give accurate recommendations with very little training. Users will deem a system useless if it cannot provide useful recommendations almost immediately.

Possible Document Features for Interestingness

There is no single feature that definitively identifies interesting articles. Pon et al. (2007) describes a set of possible aspects regarding interestingness:

1. *Topic Relevancy*: Although an article that is relevant to a topic of interest may not necessarily be interesting, relevancy to such topics is often a prerequisite for interestingness for a certain class of users. Traditional IR techniques can be used for this purpose.
2. *Uniqueness*: Articles that yield little new information compared to articles already seen may not be interesting. In contrast, an article that first breaks a news event may be interesting. Articles that describe a rare event may also be interesting. For example, Rattigan and Jensen (2005) claim that interesting articles may be produced by rare collaborations among authors. Methods for outlier detection include using mixture models (Eskin, 2000), generating solving sets (Angiulli et al., 2005) and using k-d trees (Chaudhary et al., 2002).
3. *Source Reputation*: An article's interestingness can be estimated given its source's past history in producing interesting articles. Articles from a source known to produce interesting articles tend to be more interesting than articles from less-reputable sources.
4. *Writing Style*: Most work using the writing style of articles has mainly been for authorship attribution (Koppel et al., 2006). Instead of author attribution, the same writing style features can be used to infer interestingness. For example, the vocabulary richness (Tweedie & Baayen, 1998) of an article should suit the user's understanding of the topic (e.g., a layman versus an expert). Also writing style features may help with author attribution, which can be used for source reputation, where attribution is unavailable.
5. *Freshness*: Articles about recent events tend to be labeled as more interesting than articles about older events. Also articles about the same event are published around the time the event has occurred. This may also be the case for interesting events, and consequently interesting articles.
6. *Subjectivity and Polarity*: The sentiment of an article may also contribute to a user's definition of interestingness. For example, "bad news" may be more interesting than "good news" (i.e., the polarity of the article). Or, subjective articles may be more interesting than objective articles. Polarity identification has been done with a dictionary approach (Mishne, 2005). Others have looked at subjectivity labeling, using various NLP techniques (Wiebe et al., 2004).

The above list is not an exhaustive list of interestingness features. There is currently ongoing work on

the identification and the measurement of new features that correlate with interestingness.

Ensembles

Because of the complexity of the problem of recommending articles, a solution to this problem could leverage multiple existing techniques to build a better recommendation system. In other problems, this approach has worked well, such as in webpage duplication (Henzinger, 2006).

One ensemble approach to ranking items, such as articles, is to combine multiple ranking functions through probabilistic latent query analysis (Yan & Hauptmann, 2006). Another approach uses a weighted majority algorithm to aggregate expert advice from an ensemble of classifiers to address concept drift in real-time ranking (Beckier & Arias, 2007). A simpler ensemble approach is taken by Pon et al. (2007a). Different techniques, which are relevant to determining the "interestingness" of an article, are combined together as individual features for a naïve Bayesian classifier. Pon et al. show that this achieves a better "interestingness" judgment. However, naïve Bayesian classifiers assume that features are independent. As discussed earlier, "interestingness" is complex and allows for the possibility of conditionally dependent features. For example, an article may be interesting if it is unique but relevant to topics of interest. The search for an updateable yet efficient and complete classifier for "interestingness" remains open.

Additionally, because the definition of interestingness varies from user to user (Pon et al., 2007a) and may even change over time, it is not possible to use traditional offline feature selection algorithms, such as the ones described by Guyon and Eliseeff (2003), to identify which features are important before deploying the system. So, all features are included for classification. The ideal approach to dealing with this problem is by embedding a feature selection algorithm within an updateable classifier. Some approaches have included using Winnow (Carvalho & Cohen 2006), but lack the generality for handling features with different semantic meanings. Utgoff et al.'s (1997) incremental decision tree algorithm addresses this problem but is not appropriate for an online environment due to its growing storage requirements. A different approach taken by Nurmi and Floreen (2005) identify and remove redundant features using the properties of time series data. However, this approach is not applicable to articles as articles are not necessarily dependent upon the article that immediately precedes it in the document stream.

FUTURE TRENDS

With the advent of blogs that specialize in niche news markets, readers can expect to see an explosive growth on the availability of information where only a small fraction may be of interest to them. In contrast to traditional news sources, such as CNN, blogs focus on specific topics that may be of interest to only a handful of users as opposed to the general public. This phenomenon is often referred to as the long tail market phenomenon (Anderson, 2007). Instead of building news filters that cater to the mass public, future research will focus more on personalized news recommendation. Personalization research is also present in other media, as evident in the Netflix Prize competition (2007) and the related KDD Cup 2007 competition (Bennett et al., 2007), in which teams compete to improve the accuracy of movie recommendations.

Traditional corpora, such as the ones used in TREC, are ill equipped to address the problems in personalized news recommendation. Current corpora address the traditional problems of topic relevancy and do not address the problem of interestingness. Furthermore, such corpora are not user-focused. At best, such corpora label articles that a general audience would find to be interesting as opposed to a specific user. Even the Yahoo! news articles used by Pon et al. (2007) address the problem of identifying interesting articles to a large community of users instead of a specific user. Further research in personalized news recommendation will need to be evaluated on a large test data collection that has been collected using many individual users. Such data can be collected by tracking individual user behavior on the Internet or on news bulletin boards, such as Digg (2007).

CONCLUSION

The online recommendation of interesting articles for a specific user is a complex problem, having to draw from many areas of machine learning, such as feature selection, classification, and anomaly detection. There

is no single technique that will be able to address the problem of interestingness by itself. An ensemble of multiple techniques is one possible solution to addressing this problem. Because of the growth of research in recommendation systems, more user-focused test collections should be made available for system evaluation and comparison.

REFERENCES

Anderson, C. (2007). Calculating latent demand in the long tail. *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York: ACM Press.

Angiulli, F., Basta, S., & Pizzuti, C. (2005). Detection and prediction of distance-based outliers. *Proceedings of the 2005 ACM Symposium on Applied Computing*, pages 537–542. New York: ACM Press.

Becker, H. & Arias, M. (2007). Real-time ranking with concept drift using expert advice. *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 86–94. New York: ACM Press.

Bennett, J., Elkan, C., Liu, B., Smyth, P., & Tikk, D., editors (2007). *Proceedings of KDD Cup and Workshop 2007*. ACM SIGKDD.

Carreira, R., Crato, J. M., Goncalves, D., & Jorge, J. A. (2004). Evaluating adaptive user profiles for news classification. *Proceedings of the 9th International Conference on Intelligent User Interface*, pages 206–212. New York: ACM Press.

Carvalho, V. R. & Cohen, W. W. (2006). Singlepass online learning: Performance, voting schemes and online feature selection. *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 548–553. New York: ACM Press.

Chaudhary, A., Szalay, A. S., & Moore, A. W. (2002). Very fast outlier detection in large multidimensional data sets. In *DMKD*.

Corso, G. M. D., Gulli, A., & Romani, F. (2005). Ranking a stream of news. *Proceedings of the 14th International Conference on World Wide Web*, pages 97–106. New York: ACM Press.

Digg (2007). *Digg*. Retrieved September 21, 2007, from http://www.digg.com

Eskin, E. (2000). Detecting errors within a corpus using anomaly detection. *Proceedings of the 1st Conference on North American Chapter of the Association for Computational Linguistics*, pages 148–153. San Francisco: Morgan Kaufmann Publishers Inc.

Forman, G. (2004). A pitfall and solution in multi-class feature selection for text classification. *Proceedings of the 21st International Conference on Machine Learning*, page 38.

Guyon, I. & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, *3*, 1157–1182.

Henzinger, M. (2006). Finding near-duplicate web pages: a large-scale evaluation of algorithms. *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 284–291. New York: ACM Press.

Koppel, M., Schler, J., Argamon, S., & Messeri, E. (2006). Authorship attribution with thousands of candidate authors. *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 659–660. New York: ACM Press.

Macskassy, S. A. & Provost, F. (2001). Intelligent information triage. *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 318–326. New York: ACM Press.

Mishne, G. (2005). Experiments with mood classification in blog posts. *Style2005 - the 1st Workshop on Stylistic Analysis Of Text For Information Access, at SIGIR 2005*.

Netflix (2007). *Netflix prize*. Retrieved September 21, 2007, from http://www.netflixprize.com/

Newman, D., Chemudugunta, C., Smyth, P., & Steyvers, M. (2006). Analyzing entities and topics in news articles using statistical topic models. *IEEE International Conference on Intelligence and Security Informatics*.

Nurmi, P. and Floreen, P. (2005). Online feature selection for contextual time series data. *PASCAL Subspace, Latent Structure and Feature Selection Workshop*, Bohinj, Slovenia.

Peng, F., Schuurmans, D., & Wang, S. (2003). Language and task independent text categorization with simple language models. *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 110–117. Morristown, NJ: Association for Computational Linguistics.

Pon, R. K., Cardenas, A. F., Buttler, D., & Critchlow, T. (2007a). iScore: Measuring the interestingness of articles in a limited user environment. *IEEE Symposium on Computational Intelligence and Data Mining 2007*, Honolulu, HI.

Pon, R. K., Cardenas, A. F., Buttler, D., & Critchlow, T. (2007b). Tracking multiple topics for finding interesting articles. *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 560–569. New York: ACM Press.

Radev, D., Fan, W., & Zhang, Z. (2001). Webinessence: A personalised web-based multi-document summarisation and recommendation system. *Proceedings of the NAACL-01*, pages 79–88.

Rattigan, M. & Jensen, D. (2005). The case for anomalous link detection. *4th Multi-Relational Data Mining Workshop, 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Robertson, S. & Soboroff, I. (2002). The TREC 2002 filtering track report. In *TREC11*.

Rocchio, J. (1971). *Relevance feedback in information retrieval*, pages 313–323. Prentice-Hall.

Steyvers, M. (2006). *Latent semantic analysis: A road to meaning*. Laurence Erlbaum.

Tweedie, F. J. & Baayen, R. H. (1998). How variable may a constant be? Measures of lexical richness in perspective. *Computers and the Humanities, 32*, 323–352.

Utgoff, P. E., Berkman, N. C., & Clouse, J. A. (1997). Decision tree induction based on efficient tree restructuring. *Machine Learning*, *29*(1).

Wang, J., de Vries, A. P., & Reinders, M. J. T. (2006). Unifying user-based and item-based collaborative filtering approaches by similarity fusion. *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 501–508. New York: ACM Press.

Wiebe, J., Wilson, T., Bruce, R., Bell, M., & Martin, M. (2004). Learning subjective language. *Computational Linguistics*, 30(3):277–308.

Yan, R. & Hauptmann, A. G. (2006). Probabilistic latent query analysis for combining multiple retrieval sources. *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 324–331. New York: ACM Press.

KEY TERMS

Conditionally Dependent Features: Features whose values are dependent upon the values of other features.

Ensemble: The combination of multiple techniques to achieve better results for a common task.

General Audience Recommendation: Recommendations made for the mass public, usually related to what is popular.

Interestingness: How interesting the referred item is to a specific user. This measure is complex and subjective, varying from user to user.

Online Feature Selection: The problem of selecting a subset of useful features from a set of given features for online classification by machine learning algorithms. As instances are classified sequentially, the appropriate set of features is selected for classifying each instance.

Online News Recommendation: The problem of recommending news articles to a specific user by machine learning algorithms. Such algorithms must provide a recommendation for an article when it arrives in a document stream in real-time. Once a decision on an article is made, the decision cannot be changed.

User-Focused Recommendation: Recommendations made for a specific user or a niche community.

Metaheuristics in Data Mining

Miguel García Torres
Universidad de La Laguna, Spain

Belén Melián Batista
Universidad de La Laguna, Spain

José A. Moreno Pérez
Universidad de La Laguna, Spain

José Marcos Moreno-Vega
Universidad de La Laguna, Spain

INTRODUCTION

The *Metaheuristics* are general strategies for designing heuristic procedures with high performance. The term metaheuristic, which appeared in 1986 for the first time (Glover, 1986), is compound by the terms: "meta", that means over or behind, and "heuristic". Heuristic is the qualifying used for methods of solving optimization problems that are obtained from the intuition, expertise or general knowledge (Michalewicz & Fogel, 2000).

Nowadays a lot of known strategies can be classified as metaheuristics and there are a clear increasing number of research papers and applications that use this kind of methods. Several optimization methods that already existed when the term appeared have been later interpreted as metaheuristics (Glover & Kochenberger, 2003). Genetic Algorithms, Neural Networks, Local Searches, and Simulated Annealing are some of those classical metaheuristics. Several modern metaheuristics have succeeded in solving relevant optimization problems in industry, business and engineering. The most relevant among them are Tabu Search, Variable Neighbourhood Search and GRASP. New population based evolutionary metaheuristics such as Scatter Search and Estimation Distribution Algorithms are also quite important. Besides Neural Networks and Genetic Algorithms, other nature-inspired metaheuristics such as Ant Colony Optimization and Particle Swarm Optimization are also now well known metaheuristics..

BACKGROUND

The ***Metaheuristic*** methods are general strategies for designing heuristic procedures for solving an optimization problem. An optimization problem is characterized by a search space S of feasible solutions and an objective function f. Solving the problem consists of finding an *optimal* solution s^*; i.e., a feasible solution that optimizes f in S. Given a set of transformations or moves on the solution space, the *neighbourhood* of s, denoted by $N(s)$, is the set of solutions that are reachable from s with one of these moves. A *local optimum* is a solution s that optimizes f in its neighbourhood $N(s)$. A *Local Search* is a procedure that iteratively applies an improving move to a solution (Pirlot, 1996; Yagiura & Ibaraki, 2002). The main objection to local searches is that they are trapped in a local optimum. The first metaheuristics arose looking for ways to escape from local optima in order to reach an optimal solution. There are an increasing number of books and reviews on the whole field of Metaheuristics (Reeves, 1993, Michalewicz & Fogel, 2000; Glover & Kochenberger, 2003; Blum & Roli, 2003)

Data mining (DM) is a constantly growing area. DM tools are confronted to a particular problem: the great number of characteristics that qualify data samples. They are more or less victims of the abundance of information. DM needs benefits from the powerful metaheuristics that can deal with huge amounts of data in Decision Making contexts. Several relevant tasks in DM; such as clustering, classification, feature selection and data reduction, are formulated as optimization

problems. The solutions for the corresponding problem consist of the values for the parameters that specify the role designed for performing the task. In nearest-neighbour clustering and classification, the solutions consist of the possible selections of cases for applying the rule. The objective functions are the corresponding performance measures. In Feature Selection and Data Reduction, the solutions are set of variables or cases and, if the size of set of features or the amount of data is fixed, the objective is to maximize the (predictive) performance. However in general, there are, at least, two objectives: the accuracy and the simplicity. They are usually contradictory and generally referred by the performance and the amount of information used for prediction. The accuracy is to be maximized and the amount of information is to be minimized. Therefore, multi-objective metaheuristics are appropriated to get the adequate tradeoff.

MAIN FOCUS

The main focus in the metaheuristics field related to DM is in the application of the existing and new methods and in the desirable properties of the metaheuristics. Most metaheuristic strategies have already been applied to DM tasks but there are still open research lines to improve their usefulness.

Main Metaheuristics

The *Multi-start* considers the ways to get several initial solutions for the local searches in order to escape from local optima and to increase the probability of reaching the global optimum (Martí, 2003; Fleurent & Glover, 1999). *GRASP* (*Greedy Randomized Adaptive Search Procedures*) comprises two phases, an adaptive construction phase and a local search (Feo & Resende, 1995; Resende & Ribeiro, 2003). The distinguishing feature of *Tabu Search* (Glover, 1989, 1990, Glover & Laguna, 1997) is the use of adaptive memory and special associated problem-solving strategies. *Simulated Annealing* (Kirkpatrick et al., 1983; Vidal, 1993) is derived from a local search by allowing also, probabilistically controlled, not improving moves. *Variable Neighbourhood Search* is based on systematic changes of neighbourhoods in the search for a better solution (Mladenović & Hansen, 1997; Hansen and Mladenović, 2003). *Scatter Search* (Glover, 1998; Laguna & Martí, 2002) uses an evolving reference set, with moderate size, whose solutions are combined and improved to update the reference set with quality and dispersion criteria. *Estimation of Distribution Algorithms* (Lozano & Larrañaga, 2002) is a population-based search procedure in which a new population is iteratively obtained by sampling the probability distribution on the search space that estimates the distribution of the good solutions selected from the former population. *Ant Colony Optimization* (Dorigo & Blum, 2005; Dorigo & Di Caro, 1999; Dorigo & Stützle, 2004) is a distributed strategy where a set of agents (artificial ants) explore the solution space cooperating by means of the pheromone. *Particle Swarm Optimization* (Clerc, 2006, Kennedy & Eberhart, 1995; Eberhart & Kennedy, 1995; Kennedy & Eberhat, 2001) is an evolutionary method inspired by the social behaviour of individuals within swarms in nature where a swarm of particles fly in the virtual space of the possible solutions conducted by the inertia, memory and the attraction of the best particles.

Most metaheuristics, among other optimization techniques (Olafsson et al., 2006), have already been applied to DM, mainly to Clustering and Feature Selection Problems. For instance, *Genetic Algorithms* has been applied in (Freitas, 2002), *Tabu Search* in (Tahir et al., 2007; Sung & Jin, 2000), *Simulated Annealing* in (Debuse & Rayward-Smith, 1997, 1999), *Variable Neighbourhood Search* in (Hansen and Mladenović, 2001; Belacel et al., 2002; García-López et al., 2004a), *Scatter Search* in (García-López et al., 2004b, 2006; Pacheco, 2005), *Estimation of Distribution Algorithms* in (Inza et al., 2000, 2001), *Ant Colony Ooptimization* in (Han & Shi, 2007; Handl et al., 2006; Admane et al., 2004; Smaldon & Freitas, 2006) and *Particle Swarm Optimization* in (Correa et al., 2006; Wang et al., 2007). Applications of *Neural Networks* in DM are very well known and some review or books about modern metaheuristics in DM have also already appeared (De la Iglesia et al., 1996; Rayward-Smith, 2005; Abbass et al., 2002)

Desirable Characteristics

Most authors in the field have used some of desirable properties of metaheuristics to analyse the proposed methods and few of them collected a selected list of them (Melián et al., 2003). The desirable characteristics

of the metaheuristics, from the theoretical and practical points of view, provide ways for the improvement in the field. A list of them follows:

1. ***Simplicity***: the metaheuristics should be based on a simple and clear principle, easy to understand.
2. ***Precise***: the steps or phases of the metaheuristic must be stated in precise terms, without room for the ambiguity.
3. ***Coherence***: the steps of the algorithms for particular problems should follow naturally from the principles of the metaheuristic;
4. ***Efficiency***: the procedures for particular problems should provide good solutions (optimal or near-optimal) in moderate computational time;
5. ***Efficacy***: the algorithms should solve optimally most problems of benchmarks, when available;
6. ***Effectiveness***: the procedures for particular problems should provide optimal or near-optimal solutions for most realistic instances.
7. ***Robustness***: the metaheuristics should have good performance for a variety of instances, i.e., not just be fine-tuned to some data and less good elsewhere;
8. ***Generality***: the metaheuristics should lead to good heuristics for a large variety of problems.
9. ***Adaptable***: the metaheuristics should include elements to adapt to several contexts or field of applications or different kind of models
10. ***User-friendliness***: the metaheuristics should be easy to use; without parameters or such they are easily understood and tuned.
11. ***Innovation***: the principles of the metaheuristics, and/or their use, should lead to new types of applications.
12. ***Interactivity***: the metaheuristics should allow the user to incorporate his knowledge in order to improve the performance of the procedure.
13. ***Multiplicity***: the methods should be able to present several near optimal solutions among which the user can choose.
14. ***Autonomous***: the metaheuristics should allow implementations without parameters or such that they are automatically tuned.
15. ***Applicability***: the metaheuristics should be widely applicable to several fields.

FUTURE TRENDS

Several of the future trends in metaheuristics will have a big impact in DM because they incorporate the methodologies of intelligent systems to solve the difficulties for efficiently dealing with high amount of data.

Hybrid Metaheuristics

Metaheuristics can get the benefits from the hybridization methodologies (Almeida et al., 2006; Talbi, 2002). This new emerging field includes combinations of components from different metaheuristics, low-level and high-level hybridization, portfolio techniques, expert systems, co-operative search and co-evolution techniques.

Cooperative Metaheuristics

The cooperative metaheuristics consider several search agents that implement metaheuristics and cooperate by interchanging information on the search. The cooperative scheme can be centralized or decentralized depending on the existence of a control of the communications and the way the agents use the information. They are usually obtained from the population search strategies where each individual gets search capabilities and communicates with other individuals in the population (García-Pedrajas et al., 2001; Huang, 2006; Grundel et al., 2004; Melián-Batista et al., 2006).

The Learning Metaheuristics

The performance of the metaheuristics improves by using Machine Learning tools that incorporate the knowledge obtained by the algorithm while it runs. These tools should allow a metaheuristic to tune their parameters to get the best possible performance on a set of instances (Cadenas et al., 2006; Guo, 2003).

Nature-Inspired Metaheuristics

The number and success of the metaheuristics derived from the studies of some natural phenomena are increasing in the last years. In addition to the classical *Neural Networks* and *Genetic Algorithms* other metaheuristic strategies are being consolidated in the field such as

Ant Colony Optimization and *Particle Swarm Optimization*. However recent proposals like *Artificial Immune Systems* (Timmis, 2006), *Membrane Systems* (Paun, 2002) or *Swarm Intelligence* (Engelbrecht, 2005) have not been enough studied (Forbes, 2005).

Multiple Objective Metaheuristics

Since most of the real problems in DM are multiobjective problems (where several contradictory objective functions involved), the adaptation or capabilities of metaheuristics for these problems are very relevant in real applications (Baños et al., 2006).

Parallel Metaheuristics

With the proliferation of parallel computers and faster community networks, parallel metaheuristics (Alba, 2005) is already being now an effective alternative to speed up metaheuristics searches in DM and allow efficiently dealing with high amount of data.

CONCLUSIONS

Modern metaheuristics have succeeded in solving optimization problems in relevant fields. Several relevant tasks in DM are formulated as optimization problems. Therefore, metaheuristics should provide promising tools for improving DM tasks. Some examples of this already have appeared in the specialized literature. In order to extend this success, the knowledge of the good characteristics of the successful metaheuristic is important. The relevance of the future trends of the field for DM applications depends of this knowledge.

REFERENCES

Abbass, H.A., Newton, C.S. & Sarker, R. (Eds.) (2002) *Data Mining: A Heuristic Approach.* Idea Group Publishing

Admane, L., Benatchba, K., Koudil, M., Drias, M., Gharout, S. & Hamani, N. (2004) Using ant colonies to solve data-mining problems. *IEEE International Conference on Systems, Man and Cybernetics*, 4 (10-13), 3151–3157

Alba, E. (ed.) (2005) Parallel Metaheuristics. A New Class of Algorithms. Wiley.

Almeida, F., Blesa Aguilera, M.J., Blum, C., Moreno Vega, J.M., Pérez Pérez, M., Roli, A., Sampels, M. (Eds.) (2006) Hybrid Metaheuristics. LNCS 4030, pp. 82-93, Springer

Baños, R., Gil, C., Paechter, B., Ortega, J. (2007) A Hybrid Meta-heuristic for Multi-objective Optimization: MOSATS. Journal of Mathematical Modelling and Algorithms, 6/2, 213-230.

Belacel, N., Hansen, P. & Mladenovic, N. (2002) Fuzzy J-means: a new heuristic for fuzzy clustering. *Pattern Recognition*, 35(10), 2193-2200

Blum, C. & Roli. A. (2003) Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, 35(3), 268-308

Cadenas, J.M., Canós, M.J., Garrido, M.C., Liern, V., Muñoz, E., Serrano, E. (2007) Using Data Mining in a Soft-Computing based Cooperative Multi-agent system of Metaheuristics. Journal of Applied soft Computing (in press).

Clerc, M. (2006). *Particle Swarm Optimization.* ISTE.

Correa, E.S., Freitas, S.A. & Johnson, C.G. (2006) A new Discrete Particle Swarm Algorithm Applied to Attribute Selection in a Bioinformatic Data Set. *GECCO '06: Proceedings of the 8th annual* conference on Genetic and evolutionary computation, 35–42

Debuse, J.C.W. & Rayward-Smith, V.J. (1997) Feature subset selection within a simulated annealing data mining algorithm. *J. Intelligent Information Systems*, 9(1), 57-81

Dorigo, M. & Blum, C. (2005) Ant colony optimization theory: A survey. Theoretical Computer Science, 344(2-3), 243-278

Dorigo, M. & Di Caro, G. (1999) The Ant Colony Optimization Meta-Heuristic. In Corne, D., Dorigo, M. F. & Glover, F. (Eds), *New Ideas in Optimization*, McGraw-Hill, 11-32

Dorigo, M. & Stützle, T. (2004) *Ant Colony Optimization*. MIT Press

Eberhart, R.C. & Kennedy, J. (1995) A new optimizer using particle swarm theory. *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, Nagoya, Japan, 39–43.

Engelbrecht, P. (2005) *Fundamentals of Computational Swarm Intelligence*. Wiley

Feo, T.A & Resende, M.G.C. (1995) Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6, 109-133

Fleurent, C. & Glover, F. (1999) Improved constructive multistart strategies for the quadratic assignment problem using adaptive memory. *INFORMS Journal on Computing*, 11, 198-204

Forbes, N. (2005) *Imitation of life: How Biology is Inspiring Computing*. MIT Press

Freitas, A. (2002) *Data Mining and Knowledge Discovery with Evolutionary Algorithms*. Springer

García-López, F., García-Torres, M., Melián, B., Moreno-Pérez, J.A. & Moreno-Vega, J.M. (2004a) Solving feature subset selection problem by a hybrid metaheuristic. In *Proceedings of First International Workshop in Hybrid Metaheuristics at ECAI2004*, 59-69

García-López, F., García-Torres, M., Moreno-Pérez, J.A. & Moreno-Vega, J.M. (2004b) Scatter Search for the feature selection problem. *Lecture Notes in Artificial Intelligence*, 3040, 517-525

García-López, F., García-Torres, M., Melián, B., Moreno, J.A. & Moreno-Vega, J.M. (2006) Solving feature subset selection problem by a parallel scatter search. *European Journal of Operational Research*, 169, 477-489

García-Pedrajas, N., Sanz-Tapia, E., Ortiz-Boyer, D. & Cervás-Martínez, C. (Eds.) (2001) *Introducing Multi-objective Optimization in Cooperative Coevolution of Neural Networks*. LNCS, 2084.

Glover, F. & Kochenberger, G. (Eds.) (2003) *Handbook on MetaHeuristics*. Kluwer

Glover, F. & Laguna , M. (1997) Tabu Search. Kluwer.

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. Computers and Operations Research, 5:533-549.

Glover, F. (1989) Tabu search. part I. ORSA Journal on Computing, 1:190-206.

Glover, F. (1990) Tabu search. part II. ORSA Journal on Computing, 2:4-32.

Glover, F. (1998) A template for scatter search and path relinking. In J.-K. Hao and E. Lutton, editors, Artificial Evolution, volume 1363, 13-54. Springer-Verlag.

Grundel, D., Murphey, R. & Pardalos, P.M. (Eds.) (2004) *Theory and Algorithms for Cooperative Systems*. Series on Computers and Operations Research, 4. World Scientific.

Guo, H. (2003) A Bayesian Approach to Automatic Algorithm Selection. IJCAI03 Workshop on AI and Automatic Computing.

Han, Y.F. & Shi, P.F. (2007) An improved ant colony algorithm for fuzzy clustering in image segmentation. *Neurocomputing*, 70, 665-671

Handl, J., Knowles, J. & Dorigo, M (2006) Ant-based clustering and topographic mapping. *Artificial Life*, 12, 35-61

Hansen, P. & Mladenović, N. (2001) J-means: A new local search heuristic for minimum sum-of-squares clustering. *Pattern Recognition*, 34(2), 405-413

Hansen, P. & Mladenovic, N. (2003) Variable Neighborhood Search. In F. Glover and G. Kochenberger (Eds.), *Handbook of Metaheuristics*, Kluwer, 145--184.

Huang, X. (2006) From Cooperative Team Playing to an Optimization Method. *NICSO*-2006 Granada.

Iglesia, B. de la, Debuse, J.C.W. & Rayward-Smith, V.J. (1996). Discovering Knowledge in Commercial Databases Using Modern Heuristic Techniques. *Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining KDD96*, 44-49

Inza, I., Larrañaga, P. & Sierra, B. (2000) Feature subset selection by bayesian network based optimization. *Artificial Intelligence,* 123, 157-184

Inza, I., Larrañaga, P. & Sierra, B. (2001) Feature subset selection by bayesian networks: a comparison with genetic and sequential algorithms. *International Journal of Approximate Reasoning*, 27(2), 143-164

Kennedy, J. & Eberhart, R. (1995) Particle swarm optimization. *Proceedings of the IEEE International Conference on Neural Networks*, IV, 1942–1948.

Kennedy, J. & Eberhart, R. (2001) *Swarm Intelligence*. Morgan Kaufmann.

Kirpatrick, S., Gelatt, C.D. & Vecchi, M.P. (1983) Optimization by Simulated Annealing. *Science*, 220, 671-679

Laguna, M. & Martí, R. (2002) Scatter Search Methodology and Implementations in C. Kluwer.

Lozano, J.A. & Larrañaga, P. (2002). Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation. Kluwer.

Martí, R. (2003) Multistart Methods. In *Handbook on MetaHeuristics*, Glover, F. & Kochenberger, G. (Eds.) Kluwer, 355-368

Mladenovic, N. & Hansen, P. (1997) Variable neighborhood search. *Computers and Operations Research*, 24, 1097-1100.

Melián-Batista, B., Moreno-Pérez, J.A. & Moreno Vega, J.M. (2006) Nature-inspired Decentralized Cooperative Metaheuristic Strategies for Logistic Problems. *NiSIS*-2006, Tenerife.

Michalewicz, Z. & Fogel, D.B. (2000). How to Solve It: Modern Heuristics. Springer.

Olafsson, S., Lia, X. & Wua, S. (2006) Operations research and data mining. *European Journal of Operational Research*, (doi:10.1016/j.ejor.2006.09.023), to appear

Paun, G. (2002) *Membrane Computing. An Introduction*. Springer

Pacheco, J. (2005) A Scatter Search Approach for the Minimum Sum-of-Squares Clustering Problem. *Computers and Operations Research*, 32, 1325-1335

Pirlot, M. (1996) General local search methods. European Journal of Operational Research, 92(3):493-511.

Rayward-Smith, V.J. (2005) Metaheuristics for clustering in KDD. *Evolutionary Computation*, 3, 2380-2387

Reeves, C.R. (ed.) (1993) Modern Heuristic Techniques for Combinatorial Problems. Blackwell.

Resende, M.G.C. & Ribeiro, C.C. (2003) Greedy randomized adaptive search procedures. In Glover, F. & Kochenberger, G. (Eds.) *Handbook of Metaheuristics*, Kluwer, 219-249

Smaldon, J. & Freitas, A. (2006) A new version of the ant-miner algorithm discovering unordered rule sets. *Proceedings of GECCO '06*, 43-50

Sung, C.S. & Jin, H.W. (2000) A tabu-search-based heuristic for clustering. *Pattern Recognition*, 33, 849-858

Talbi, E-G. (2002) A Taxonomy of Hybrid Metaheuristics, Journal of Heuristics 8(5), 541-564

Tahir, M.A., Bouridane, A. & Kurugollu, F. (2007) Simultaneous feature selection and feature weighting using Hybrid Tabu Search/K-nearest neighbor classifier. *Pattern Recognition Letters*, 28, 438-446

Timmis, J. (2006). Artificial Immune Systems - Today and Tomorrow. *To appear in Special Issue on Natural and Artificial Immune Systems. Natural Computation.*

Yagiura M. and Ibaraki T. (2002) Local search. In P.M. Pardalos and M.G.C. Resende, (Eds.), Handbook of Applied Optimization, 104-123. Oxford University Press.

Vidal, R.V.V. (1993) Applied simulated annealing. Springer, 1993.

Wang, X.Y., Yang, J., Teng, X.L., Xia, W.J. & Jensen, R. (2007) Feature selection based on rough sets and particle swarm optimization. *Pattern Recognition Letters*, 28, 459-471

KEY TERMS

Diversification: The capability of the search methods to explore different zones of the search space.

Heuristic: A procedure to provide a good solution of an optimization problem that is obtained from the intuition, expertise or general knowledge.

Intensification: The capability of the search methods for improving the solutions of an optimization problem.

Local Optimum: A solution of an optimization problem that is better that any other solution of its neighbourhood.

Local Search: A heuristic method consisting of iteratively applying an improving move to a solution until a stopping criterion is met.

Memory: The capability of some search methods for using the information on the solutions examined in the search process and their objective values.

Metaheuristics: The metaheuristics are general strategies for designing heuristic procedures with high performance.

Neighbourhood: Given a set of moves or transformations in the solution space of an optimization problem, the neighbourhood of a solution is the set of solutions that can be obtained from it by one of these moves or transformations.

Optimization Problem: Given a set S of alternative solutions and an objective function *f*, the corresponding optimization problem consists in finding the solution *s* that optimizes *f*.

Population Method: A solution method, based on a set of solutions (the population), that searches for the optimum in the solution space of an optimization problem.

Meta-Learning

Christophe Giraud-Carrier
Brigham Young University, USA

Pavel Brazdil
University of Porto, Portugal

Carlos Soares
University of Porto, Portugal

Ricardo Vilalta
University of Houston, USA

INTRODUCTION

The application of Machine Learning (ML) and Data Mining (DM) tools to classification and regression tasks has become a standard, not only in research but also in administrative agencies, commerce and industry (e.g., finance, medicine, engineering). Unfortunately, due in part to the number of available techniques and the overall complexity of the process, users facing a new data mining task must generally either resort to trial-and-error or consultation of experts. Clearly, neither solution is completely satisfactory for the non-expert end-users who wish to access the technology more directly and cost-effectively.

What is needed is an informed search process to reduce the amount of experimentation with different techniques while avoiding the pitfalls of local optima that may result from low quality models. Informed search requires meta-knowledge, that is, knowledge about the performance of those techniques. Meta-learning provides a robust, automatic mechanism for building such meta-knowledge. One of the underlying goals of meta-learning is to understand the interaction between the mechanism of learning and the concrete contexts in which that mechanism is applicable. Meta-learning differs from base-level learning in the scope of adaptation. Whereas learning at the base-level focuses on accumulating experience on a specific learning task (e.g., credit rating, medical diagnosis, mine-rock discrimination, fraud detection, etc.), learning at the meta-level is concerned with accumulating experience on the performance of multiple applications of a learning system.

The meta-knowledge induced by meta-learning provides the means to inform decisions about the precise conditions under which a given algorithm, or sequence of algorithms, is better than others for a given task. While Data Mining software packages (e.g., SAS Enterprise Miner, SPSS Clementine, Insightful Miner, PolyAnalyst, KnowledgeStudio, Weka, Yale, Xelopes) provide user-friendly access to rich collections of algorithms, they generally offer no real decision support to non-expert end-users. Similarly, tools with emphasis on advanced visualization help users understand the data (e.g., to select adequate transformations) and the models (e.g., to tweak parameters, compare results, and focus on specific parts of the model), but treat algorithm selection as a post-processing activity driven by the users rather than the system. Data mining practitioners need systems that guide them by producing explicit advice automatically. This chapter shows how meta-learning can be leveraged to provide such advice in the context of algorithm selection.

BACKGROUND

STABB is an early precursor of meta-learning systems in the sense that it was the first to show that a learner's bias can be adjusted dynamically (Utgoff, 1986). VBMS may be viewed as the first simple meta-learning system (Rendell et al., 1989). It learns to choose one of three symbolic learning algorithms as a function of the number of training instances and the number of features. The StatLog project extended VBMS significantly by considering a larger number of

dataset characteristics, together with a broad class of candidate models and algorithms for selection (Brazdil & Henery, 1994). The aim was to characterize the space in which a given algorithm achieves positive generalization performance.

The MLT project focused on the practice of machine learning and produced a toolbox consisting of a number of learning algorithms, datasets, standards and know-how (Kodratoff et al., 1992; Craw et al., 1992). Considerable insight into many important machine learning issues was gained during the project, much of which was translated into meta-rules that formed the basis of a kind of user-guidance expert system called Consultant-2.

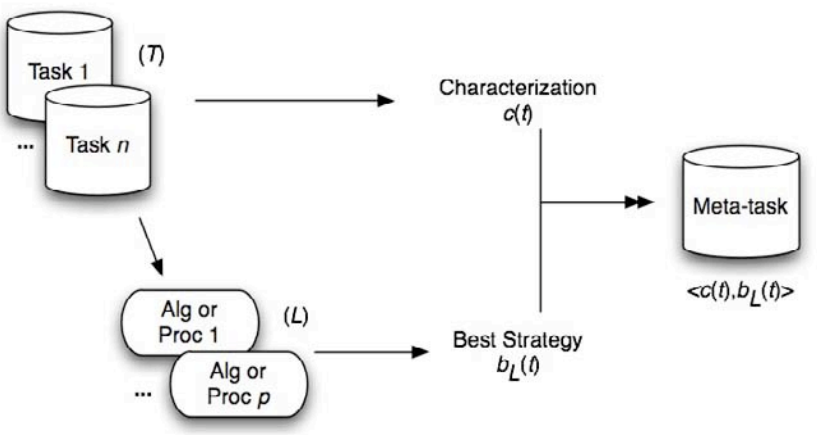
Born out of practical challenges faced by researchers at Daimler Benz AG (now), CITRUS is perhaps the first implemented system to offer user guidance for the complete data mining process, rather than for a single phase of the process (Engels, 1996; Wirth et al., 1997). Algorithm selection takes place in two stages, consisting of: 1) mapping tasks to classes of algorithms, and 2) selecting an algorithm from the selected class. The mapping stage is driven by decomposition and guided by high-level pre/post-conditions (e.g., interpretability). The selection stage consists of using data characteristics (inspired by the Statlog project) together with a process of elimination (called "strike-through"), where algorithms that do not work for the task at hand are successively eliminated until the system finds one applicable algorithm. Although there is no meta-learning in the traditional sense in CITRUS, there is still automatic guidance beyond the user's own input.

Finally, theoretical results, such as the NFL theorems and their consequences have helped in identifying limitations and opportunities for meta-learning (Schaffer, 1994; Wolpert & Macready, 1995; Wolpert, 2001). Additionally, extensive empirical studies have confirmed the theory, and provided additional insight into learning that may serve both as a source of direct meta-knowledge and as input to meta-learning (Aha, 1992; Holte, 1993; Lim et al., 2000).[1]

MAIN FOCUS

Meta-learning, in the context of model selection, consists of applying learning mechanisms to the problem of mapping learning tasks to algorithms. Let L be a set of learning algorithms and T be a set of learning tasks such that for each t in T, $b_L(t)$ represents the algorithm in L that performs best on t for some user-defined performance criterion (e.g., predictive accuracy, execution time).[2] Since learning tasks may be unwieldy to handle

Figure 1. Meta-dataset construction



directly, some type of task characterization is used and the meta-learner actually learns a mapping from characterizations to algorithms. For each learning task t, let $c(t)$ denote the characterization of t by some fixed mechanism. The set $\{<c(t), b_L(t)>: t \text{ in } T\}$ constitutes a meta-task or meta-dataset, as depicted in Figure 1.

A meta-learner can then take the meta-dataset $\{<c(t), b_L(t)>: t \text{ in } T\}$ as a training set and induce a meta-model that, for each new learning task, predicts the model from L that will perform best. Alternatively, one may build a meta-model that predicts a ranking of algorithms from L (Berrer et al., 2000; Brazdil et al., 2003). The ranking approach reduces the brittleness of the meta-model. Assume, for example, that the model predicted best for some new learning task results in what appears to be a poor performance. In the single-model prediction approach, the user has no further information as to what other model to try. In the ranking approach, the user may try the second best, third best, and so on, in an attempt to improve performance. Furthermore, ranking makes it easier to include additional (possibly qualitative) criteria, such as comprehensibility, in the selection process (Giraud-Carrier, 1998).

Clearly, one of the challenges of meta-learning is the construction of the meta-dataset, i.e, $<c(t), b_L(t)>$ pairs for some base level learning tasks. This raises issues with: 1) the choice of the characterization mechanism c, 2) the choice of the set of learners L, 3) the collection of representative tasks, and 4) the cost of computing $c(t)$ and $b_L(t)$ for each task. We briefly discuss each of these issues in the following sections.

Characterization Mechanism

As in any learning task, the characterization of the examples plays a crucial role in enabling learning. The central idea is that high-quality dataset characteristics or meta-features provide useful information to discriminate among the performances of a set of given learning strategies. Typical characterization techniques belong to one of the following classes.

- *Statistical and Information-Theoretic Characterization.* A number of statistical and information-theoretic measures are extracted from the dataset, such as number of classes, number of features, ratio of examples to features, degree of correlation between features and target concept, average class entropy and class-conditional entropy, skewness, kurtosis, signal–to-noise ratio, etc. (Aha, 1992; Michie et al., 1994; Engels & Theusinger, 1998; Sohn, 1999; Köpf et al., 2000; Kalousis, 2002).
- *Model-Based Characterization*. Models induced on the dataset are used as indicators of the underlying properties of the dataset. To date, only decision trees have been used for the extraction of characteristics such as nodes per feature, maximum tree depth, shape, tree imbalance, etc. (Bensusan et al., 2000; Peng et al., 2002).
- *Landmarking*. The performances of simple learners, known as landmarkers, are computed on the dataset using cross-validation (Bensusan & Giraud-Carrier, 2000; Pfahringer et al., 2000). The idea is that landmarkers serve as signposts of the performance of the full-fledged target learners in L. Alternatively, one can exploit accuracy results obtained on simplified versions of the data (e.g., samples), known as sub-sampling landmarks (Fürnkranz & Petrak, 2001; Soares et al., 2001).

Choice of Base-level Learners

Although no learner is universal, each learner has its own area of expertise, which can be informally defined as the set of learning tasks on which it performs well. Since the role of the meta-model is to predict which algorithm is most likely to perform best on each new task, one should select base learners with complementary areas of expertise. In principle, one should seek the smallest set of learners that is most likely to ensure a reasonable coverage of the base-level learning space. One way to ensure diversity is by choosing representative learners from varied model classes. The more varied the biases, the greater the coverage.

Meta-Dataset Construction

The number of accessible, documented, real-world learning tasks is relatively small, which poses a challenge for learning. This challenge may be addressed either by augmenting the meta-dataset through systematic generation of synthetic base level tasks, or by taking the view that the model selection task is inherently incremental and treating it as such. The second approach results in slower learning since learning tasks become available over time. On the other hand, it naturally adapts to reality, extending to new areas

of the base level learning space only when tasks from these areas actually arise.

Computational Cost

The computational cost is the price to pay to be able to perform model selection learning at the meta-level. However, in order to be justifiable, the cost of computing $c(t)$ should be significantly lower than the cost of computing $b_L(t)$. Otherwise, even if the meta-model is very accurate, it has little value as the user would be better off trying all algorithms and selecting the best one, which clearly defeats the purpose of meta-learning. The characterization mechanisms listed above all include many measures that satisfy this condition.

Although much remains to be done, results suggest the suitability of meta-learning for model selection (Brazdil & Soares, 2000; Bensusan & Kalousis, 2001; Hilario & Kalousis, 2001). We briefly describe two recent, successful systems as an illustration. One is a strict meta-learning system and offers ranking advice for model selection. The other is based on an ontology, but produces ranking advice for the complete KDD process.

Data Mining Advisor

The Data Mining Advisor (DMA) is the main product of the ESPRIT METAL research project (see http://www.metal-kdd.org). The DMA is a Web-based meta-learning system for the automatic selection of model building algorithms in the context of classification and regression tasks. Given a dataset and goals defined by the user in terms of accuracy and training time, the DMA returns a list of algorithms that are ranked according to how well they are predicted to meet the stated goals.

The DMA guides the user through a wizard-like step-by-step process consisting of 1) uploading a target dataset (with some user-defined level of privacy), 2) characterizing the dataset automatically using statistical and information-theoretic measures, 3) setting the selection criteria and the ranking method, 4) producing the ranking advice, and 5) executing user-selected algorithms on the dataset. Although the induced models themselves are not returned, the DMA reports 10-fold cross-validation accuracy, true rank and score, and, when relevant, training time. A simple example of the ranking produced by the DMA is shown in Figure 2, where some algorithms were selected for execution.

The DMA's choice of providing rankings rather than "best-in-class" is motivated by a desire to give as much information as possible to the user, as discussed above. In some sense, one can argue that the ranking approach subsumes the "best-in-class" approach. Interestingly, empirical evidence suggests that the best algorithm is generally within the top 3 in the DMA rankings (Brazdil et al., 2003).

Intelligent Discovery Assistant

The notion of Intelligent Discovery Assistant (IDA) provides a template for building ontology-driven, pro-

Figure 2. Sample DMA output

Ranking table

Dowload results | register now! | learn more about the online advisor

Predicted Rank	Algorithm	Predicted Score	Status	Run	Accuracy	Time	True Rank	True Score
1.	c50rules	1.031	finished	--	0.2830000	?	6	1.003
2.	lindiscr	1.03	finished	--	0.2340000	?	1	1.048
3.	c50tree	1.026	--		--	--	--	--
4.	ltree	1.023	--		--	--	--	--
5.	clemMLP	1.017	finished	--	0.2680000	?	5	1.006
6.	c50boost	1.017	--		--	--	--	--
7.	ripper	1.009	--		--	--	--	--
8.	mlcnb	1	finished	--	0.2430000	?	2	1.036
9.	clemRBFN	0.948	--		--	--	--	--
10.	mlcib1	0.913	finished	--	0.3180000	?	10	0.938

cess-oriented assistants for KDD (Bernstein & Provost, 2001; Bernstein et al., 2005). IDAs encompass the three main algorithmic steps of the KDD process, namely, pre-processing, model building and post-processing. In IDAs, any chain of operations consisting of one or more operations from each of these steps is called a Data Mining (DM) process. The goal of an IDA is to propose to the user a list of ranked DM processes that are both valid and congruent with user-defined preferences (e.g., speed, accuracy).

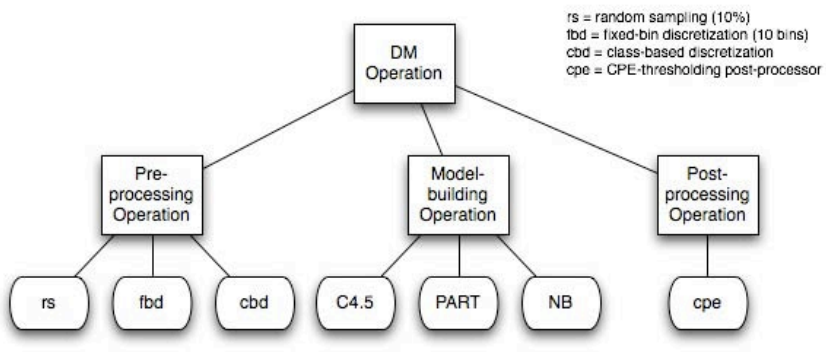
The IDA's underlying ontology is essentially a taxonomy of DM operations or algorithms, where the leaves represent implementations available in the corresponding IDA. Operations are characterized by pre-conditions, post-conditions and heuristic indicators. Clearly, the versatility of an IDA is a direct consequence of the richness of its ontology. The typical organization of an IDA consists of 1) the plan generator, that uses the ontology to build a list of (all) valid DM processes that are appropriate for the task at hand, and 2) the heuristic ranker, that orders the generated DM processes according to preferences defined by the user.

The plan generator takes as input a dataset, a user-defined objective (e.g., build a fast, comprehensible classifier) and user-supplied information about the data, i.e., information that may not be obtained automatically. Starting with an empty process, it systematically searches for an operation whose pre-conditions are met and whose indicators are congruent with the user-defined preferences. Once an operation has been found, it is added to the current process, and its post-conditions become the system's new conditions from which the search resumes. The search ends once a goal state has been found or when it is clear that no satisfactory goal state may be reached. The plan generator's search is exhaustive: all valid DM processes are computed. Figure 3 shows the output of the plan generator for a small ontology of only 7 operations, when the input dataset is continuous-valued and comprehensible classifiers are to be preferred.

The restriction of the plan generator to valid processes congruent with user-defined objectives is generally sufficient to make an exhaustive search feasible. The main advantage of this exhaustivity is that no valid DM process is ever overlooked, as is likely to be the case with most users, including experts. As a result, an IDA may, and evidence suggests that it does, uncover novel processes that experts had never thought about before, thus enriching the community's meta-knowledge (Bernstein & Provost, 2001).

Once all valid DM processes have been generated, a heuristic ranker is applied to assist the user further, by organizing processes in descending order of ``return'' on user-specified goals. For example, the processes in Figure 3 are ordered from simplest (i.e., least number of steps) to most elaborate. The ranking relies on the knowledge-based heuristic indicators. If speed rather than simplicity were the objective, for example, then

Figure 3. Sample list of IDA-generated DM processes



Plan #1: C4.5
Plan #2: PART
Plan #3: rs, C4.5
Plan #4: rs, PART
Plan #5: fbd, C4.5
Plan #6: fbd, PART
Plan #7: cbd, C4.5
Plan #8: cbd, PART
Plan #9: rs, fbd, C4.5
Plan #10: rs, fbd, PART
Plan #11: rs, cbd, C4.5
Plan #12: rs, cbd, PART
Plan #13: fbd, NB, cpe
Plan #14: cbd, NB, cpe
Plan #15: rs, fbd, NB, cpe
Plan #16: rs, cbd, NB, cpe

Plan #3 would be bumped to the top of the list, and all plans involving random sampling would also move up. In the current implementation of IDAs, rankings rely on fixed heuristic mechanisms. However, IDAs are independent of the ranking method and thus, they could possibly be improved by incorporating meta-learning to generate rankings based on past performance.

FUTURE TRENDS

One important research direction in meta-learning consists in searching for improved meta-features in the characterization of datasets. A proper characterization of datasets can elucidate the interaction between the learning mechanism and the task under analysis. Current work has only started to unveil relevant meta-features; clearly much work lies ahead. For example, many statistical and information-theoretic measures adopt a global view of the dataset under analysis; meta-features are obtained by averaging results over the entire training set, implicitly smoothing the actual distribution. There is a need for alternative and more detailed descriptors of the example distribution in a form that highlights the relationship to the learner's performance.

Similarly, further research is needed in characterizing learning algorithms. Recent efforts in model composition may prove useful. In this paradigm, instead of seeking to combine several whole algorithms or to find one algorithm among several that would perform best on a given task, the system breaks the learning process down into sub-components and, for each task, composes a custom, possibly novel, learning system from a combination of these components (Suyama et al., 1998; Abe & Yamaguchi, 2002).

Recently proposed agent-based data mining architectures offer unique ways to increase the versatility, extensibility and usability of meta-learning (Botía et al., 2001; Hernansaez et al., 2004; Zhong et al., 2001).

Finally, the increased amount and detail of data available about the operations of organizations is leading to a demand for a much larger number of models, up to hundreds or even thousands. This kind of application has been called Extreme Data Mining (Fogelman-Soulié, 2006). Current DM methodologies, which are largely dependent on human efforts, are not suitable for this kind of extreme settings because of the large amount of human resources required. Meta-learning can be used to reduce the need for human intervention in model development and thus, we expect that it will play a major role in these large-scale Data Mining applications.

CONCLUSION

From a practical standpoint, meta-learning helps solve important problems in the application of data mining tools. First, the successful use of these tools outside the boundaries of research is conditioned upon the appropriate selection of a suitable predictive model, or combination of models, according to the domain of application. Without some kind of assistance, model selection and combination can turn into solid obstacles to non-expert end-users. Second, a problem commonly observed in the practical use of data mining tools is how to profit from the repetitive use of predictive models over similar tasks. The successful application of models in real-world scenarios requires continuous adaptation to new needs. Rather than starting afresh on new tasks, one would expect the learning mechanism itself to re-learn, taking into account previous experience. Again, meta-learning systems can help control the process of exploiting cumulative expertise by searching for patterns across tasks, thus improving the utility of data mining. Interestingly, generalizations of meta-learning for algorithm selection in other areas, such as cryptography, sorting and optimization, have recently been proposed (Smith-Miles, 2007).

REFERENCES

Abe, H., & Yamaguchi, T. (2002). Constructing Inductive Applications by Meta-Learning with Method Repositories, in *Progress in Discovery Science, Final Report of the Japanese Discovery Science Project*, LNCS 2281, Springer-Verlag, 576-585.

Aha D. W. (1992). Generalizing from Case Studies: A Case Study, in *Proceedings of the Ninth International Workshop on Machine Learning*, 1-10.

Bensusan, H., & Giraud-Carrier, C. (2000). Discovering Task Neighbourhoods Through Landmark Learning Performances, in *Proceedings of the Fourth European Conference on Principles and Practice of Knowledge Discovery in Databases*, 325,330.

Bensusan H., Giraud-Carrier C., & Kennedy C.J. (2000). A Higher-Order Approach to Meta-Learning, in *Proceedings of the ECML-2000 Workshop on Meta-Learning: Building Automatic Advice Strategies for Model Selection and Method Combination*, 109-118.

Bensusan, H., & Kalousis, A. (2001). Estimating the Predictive Accuracy of a Classifier, in *Proceedings of the Twelfth European Conference on Machine Learning*, 25-36.

Bernstein, A., & Provost, F. (2001). An Intelligent Assistant for the Knowledge Discovery Process, in *Proceedings of the IJCAI-01 Workshop on Wrappers for Performance Enhancement in KDD*.

Bernstein, A., Provost, F., & Hill, S. (2005). Toward Intelligent Assistance for a Data Mining Process: An Ontology-based Approach for Cost-sensitive Classification, *IEEE Transactions on Knowledge and Data Engineering*, 17(4):503-518.

Berrer, H., Paterson, I., & Keller, J. (2000). Evaluation of Machine-learning Algorithm Ranking Advisors, in *Proceedings of the PKDD-2000 Workshop on Data-Mining, Decision Support, Meta-Learning and ILP: Forum for Practical Problem Presentation and Prospective Solutions*.

Botía, J.A., Gómez-Skarmeta, A.F., Valdés, M., & Padilla, A. (2001). METALA: A Meta-learning Architecture, in *Proceedings of the International Conference, Seventh Fuzzy Days on Computational Intelligence, Theory and Applications* (LNCS 2206), 688-698.

Brazdil, P., & Henery, B. (1994). Analysis of Results, in Michie, D. et al. (Eds.), *Machine Learning, Neural and Statistical Classification*, Ellis Horwood, Chapter 10.

Brazdil, P., & Soares, C. (2000). A Comparison of Ranking Methods for Classification Algorithm Selection, in *Proceedings of the Eleventh European Conference on Machine Learning*, 63-74.

Brazdil, P., Soares, C., & Pinto da Costa, J. (2003). Ranking Learning Algorithms: Using IBL and Meta-Learning on Accuracy and Time Results, *Machine Learning*, 50(3):251-277.

Craw, S., Sleeman, D., Granger, N., Rissakis, M., & Sharma, S. (1992). Consultant: Providing Advice for the Machine Learning Toolbox, in Bramer, M., & Milne, R. (Eds.), *Research and Development in Expert Systems IX (Proceedings of Expert Systems'92)*, SGES Publications, 5-23.

Engels, R. (1996). Planning Tasks for Knowledge Discovery in Databases; Performing Task-Oriented User-Guidance, in *Proceedings of the Second ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 170-175.

Engels, R., & Theusinger, C. (1998). Using a Data Metric for Offering Preprocessing Advice in Data-mining Applications, in *Proceedings of the Thirteenth European Conference on Artificial Intelligence*.

Fogelman-Soulié, F. (2006). Data Mining in the Real World: What Do We Need and What Do We Have? in *Proceedings of the KDD-2206 Workshop on Data Mining for Business Applications*, 44-48.

Fürnkranz, J., & Petrak J. (2001). An Evaluation of Landmarking Variants, in *Working Notes of the ECML/PKDD 2001 Workshop on Integrating Aspects of Data Mining, Decision Support and Meta-Learning*.

Giraud-Carrier, C. (1998). Beyond Predictive Accuracy: What?, in *Proceedings of the ECML-98 Workshop on Upgrading Learning to the Meta-Level: Model Selection and Data Transformation*, 78-85.

Hernansaez, J.M., Botía, J.A., & Gómez-Skarmeta, A.F. (2004). A J2EE Technology Based Distributed Software Architecture for Web Usage Mining, in *Proceedings of the Fifth International Conference on Internet Computing*, 97-101.

Hilario, M., & Kalousis, A. (2001). Fusion of Meta-Knowledge and Meta-Data for Case-Based Model Selection, in *Proceedings of the Fifth European Conference on Principles of Data Mining and Knowledge Discovery*, 180-191.

Holte, R.C. (1993). Very Simple Classification Rules Perform Well on Most Commonly Used Datasets, *Machine Learning*, 11:63-91.

Kalousis, A. (2002). Algorithm Selection via Meta-Learning, Ph.D. Thesis, University of Geneva, Department of Computer Science.

Kodratoff, Y., Sleeman, D., Uszynski, M., Causse, K., & Craw, S. (1992). Building a Machine Learning Toolbox, in Steels, L., & Lepape, B. (Eds.), *Enhancing the Knowledge Engineering Process*, Elsevier Science Publishers, 81-108.

Köpf, C., Taylor, C.C., & Keller, J. (2000). Meta-analysis: From Data Characterisation for Meta-learning to Meta-regression, in *Proceedings of the PKDD-2000 Workshop on Data Mining, Decision Support, Meta-learning and ILP: Forum for Practical Problem Presentation and Prospective Solutions*.

Lim, T-S., Loh, W-Y., & Shih, Y-S. (2000). A Comparison of Prediction Accuracy, Complexity and Training Time of Thirty-Three Old and New Classification Algorithms, *Machine Learning*, 40:203-228.

Michie, D., Spiegelhalter, D.J., & Taylor, C.C. (1994). *Machine Learning, Neural and Statistical Classification*, England: Ellis Horwood.

Peng, Y., Flach, P., Brazdil, P., & Soares, C. (2002). Decision Tree-Based Characterization for Meta-Learning, in *Proceedings of the ECML/PKDD'02 Workshop on Integration and Collaboration Aspects of Data Mining, Decision Support and Meta-Learning*, 111-122.

Pfahringer, B., Bensusan, H., & Giraud-Carrier, C. (2000). Meta-learning by Landmarking Various Learning Algorithms, in *Proceedings of the Seventeenth International Conference on Machine Learning*, 743-750.

Rendell, L., Seshu, R., & Tcheng, D. (1989). Layered Concept Learning and Dynamical Variable Bias Management, in *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*.

Schaffer, C. (1994). A Conservation Law for Generalization Performance, in *Proceedings of the Eleventh International Conference on Machine Learning*, 259-265.

Smith-Miles, K. (2007). Cross-disciplinary Perspectives on the Algorithm Selection Problem: Recent Advances and Emerging Opportunities, Technical Report TR C07/07, Deakin University, School of Engineering and Information Technology.

Soares, C., Petrak, J., & Brazdil, P. (2001). Sampling-Based Relative Landmarks: Systematically Test-Driving Algorithms before Choosing, in *Proceedings of the Tenth Portuguese Conference on Artificial Intelligence*.

Sohn, S.Y. (1999). Meta Analysis of Classification Algorithms for Pattern Recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence, 21(11):1137-1144.

Suyama, A., Negishi, N., & Yamaguchi, T. (1998). CAMLET: A Platform for Automatic Composition of Inductive Learning Systems Using Ontologies, in *Proceedings of the Fifth Pacific Rim International Conference on Artificial Intelligence*, 205-215.

Utgoff, P.E. (1986). Shift of Bias for Inductive Concept Learning, in Michalski, R.S., Carbonell, J.G., & Mitchell, T.M. (Eds.), *Machine Learning: An Artificial Intelligence Approach, Volume II*, Morgan Kaufmann Publishers. Inc., Chapter 5.

van Someren, M. (2001). Model Class Selection and Construction: Beyond the Procustean Approach to Machine Learning Applications, in Paliouras, G., Karkaletsis, V., & Spyropoulos, C.D. (Eds.), *Machine Learning and Its Applications: Advanced Lectures, LNCS 2049*, Springer-Verlag, 196-217.

Wirth, R., Shearer, C., Grimmer, U., Reinartz, T.P., Schlosser, J., Breitner, C., Engels, R., & Lindner, G. (1997). Towards Process-Oriented Tool Support for Knowledge Discovery in Databases, in Proceedings of the First European Conference on Principles and Practice of Knowledge Discovery in Databases, 243-253.

Wolpert, D.H., & Macready, W.G. (1995). No Free Lunch Theorems for Search, Technical Report SFI-TR-95-02-010, Santa Fe Institute.

Wolpert, D.H. (2001). The Supervised Learning No-Free-Lunch Theorems, in *Proceedings of the Sixth Online World Conference on Soft Computing in Industrial Applications*, 325-330.

Zhong, N., Liu, C., & Oshuga, S. (2001). Dynamically Organizing KDD Processes, *International Journal of Pattern Recognition and Artificial Intelligence*, 15(3):451-473.

KEY TERMS

Data Mining: Application of visualization, statistics and machine learning to the discovery of patterns in databases. There is general consensus that patterns found by data mining should in some way be novel and actionable.

Landmarking: A task characterization that replaces a learning task by the performances of a number of simple and efficient learning algorithms on that task.

Meta-Dataset: Dataset consisting of task characterizations (or meta-features) together with their associated best strategy (i.e., learning algorithm or data mining process that gives the best performance on the task).

Meta-Features: Features used to characterize datasets, that serve as inputs to meta-learning. These features may take the form of statistics, landmarkers or model-based attributes.

Meta-Learning: Application of learning techniques at the meta-level. Any use of learning methods to help inform the process of machine learning. Learning about learning.

Task Characterization: A method for extracting features, then known as meta-features, from the dataset associated with a learning task.

ENDNOTES

[1] Note that, although it is sometimes viewed as a form of meta-learning, we purposely omit from this discussion the notion of model combination. Model combination consists of creating a single learning system from a collection of learning algorithms. It has been shown that in many cases improved performance is obtained by combining the strengths of several learning algorithms. These approaches reduce the probability of misclassification based on any single induced model by increasing the system's area of expertise through combination. However, from the meta-learning perspective, they can be regarded as single algorithms.

[2] It is easy to extend the notion of best learning algorithm for *t* to best data mining process for *t*.

A Method of Recognizing Entity and Relation

Xinghua Fan
Chongqing University of Posts and Telecommunications, China

INTRODUCTION

Entity and relation recognition, i.e. assigning semantic classes (e.g., person, organization and location) to entities in a given sentence and determining the relations (e.g., born-in and employee-of) that hold between the corresponding entities, is an important task in areas such as information extraction (IE) (Califf and Mooney, 1999; Chinchor, 1998; Freitag, 2000; Roth and Yih, 2001), question answering (QA) (Voorhees, 2000; Changki Lee et al., 2007) and story comprehension (Hirschman et al., 1999). In a QA system, many questions ask for the specific entities involved in some relations. For example, the question that "Where was Poe born?" in TREC-9 asks for the location entity in which Poe was born. In a typical IE extraction task such as constructing a jobs database from unstructured text, the system has to extract many meaning entities like title and salary, ideally, to determine whether the entities are associated with the same position.

BACKGROUND

In all earlier works as we know of, except for Roth and Yih' work (2002), the entity and relation recognition task is treated as two separate subtasks, which are typically carried out sequentially, i.e. firstly, use an entity recognizer to identify entities, and then use a relation classifier to determine the relations. This procedure is problematic: firstly, errors made by the entity recognizer are propagated to the relation classifier with an accumulative effect and may degrade its performance significantly. For example, if "Boston" is mislabeled as a person, it will never be classified as the location of Poe's birthplace. Secondly, some relation information can be available only during the relation recognition, and the information is sometimes crucial to resolve ambiguity of entity recognition. For example, if we know that the class of an entity corresponds to the first argument X in the relation born-in (X, China), the class of the entity cannot be a location but a person.

To resolve the first problem described above, Roth and Yih (2002) developed a method, in which the two subtasks are carried out simultaneously. Firstly, two classifiers are trained for entities and relations independently, and their outputs are treated as the conditional probability distributions for each entity and relation. Secondly, this information together with the constraint knowledge induced among relations and entities are represented in a belief network (Pearl, 1998) and used to make global inferences for all entities and relations of interest. The idea of making global inferences is very good because it blocks the errors propagation. But the approach has still several problems. Firstly, the belief network cannot fully model the entity and relation recognition question since it allows no cycles, and the question need a probability model that can deal with loop. For example, relation R_{12} and R_{21} actually depend on each other (e.g., if R_{12} is born-in, then R_{21} will not be born-in), so there exists a loop between entity E_1 and E_2 through R_{12} and R_{21}. Secondly, the method cannot use some constraint knowledge that is only available during relation recognition and helpful to improve the precision of entity recognition.

To overcome the shortages described above, Fan and Sun (2005, 2006) presented a solution, which includes (1) to present the model dubbed "entity relation propagation diagram" and "entity relation propagation tree", and (2) to develop a method for this task based on the model. The presented method allows subtasks entity recognition and relation recognition to be linked more closely together. At the same time, the method can model the loopy dependencies among entities and relations, and can use two kinds of constraint knowledge learned from the annotated dataset.

MAIN FOCUS

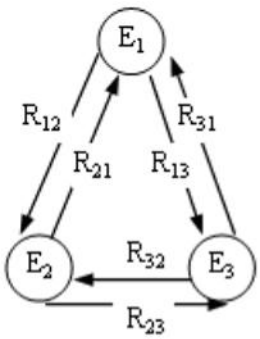
The Problem of Entity and Relation Recognition

For readability, an example sentence is illustrated as Figure 1. Conceptually, the entities and relations in a sentence can be viewed, while taking account of the mutual dependencies among them, as the labeled graph in Figure 2. In Figure 2(a), a node represents an entity and a link denotes the relation held between two entities. The arrowhead of a link represents the direction of the relation. Each entity or relation has several attributes respectively, which are denoted as a corresponding table (Figure 2(b) and Figure 2(c)). These attributes can be classified into two classes. Some of them that are easy to acquire via learned classifiers, such as words inside an entity and parts of speech of words in the context, are called local attribute; and other attributes that are difficult to acquire, such as semantic classes of phrases and relations among them, are called decision attribute. The question of recognizing entity and relation is to determine a unique value for each decision attribute of all entities and relations, while taking account of the local attributes of all entities and all relations. To describe the problem in a formal way, some basic definitions are given as follows.

Figure 1. A sentence that has three entities

Dole (E_1)'s wife, Elizabeth (E_2), is a native of Salisbury, N.C. (E_3)

Figure 2. Conceptual view of entity and relation



(a)

Entity Name	E_1	E_2	E_3
Attribute-1 (words)	Dole	Elizabeth	Salisbury, N.C.
Attribute-2 (POS)	Noun	Noun	Noun
…	…	…	…
Attribute-m	…	…	…

(b)

Relation Name	R_{12}	R_{23}
Attribute-1 (special pattern)	E_1's wife, E_2	E_2 be naïve of E_3
…	…	…
Attribute-n	…	…

(c)

Definition 1 (Entity): An entity can be a single word or a set of consecutive words with a predefined boundary. A sentence is a linked list, which consists of words and entities. Entities in a sentence are denoted as E_1, E_2 … according to their order, and they take values that range over a set of entity class C^E. For example, the sentence in Figure 1 has three entities: E_1= "Dole", E_2= "Elizabeth" and E_3= "Salisbury, N.C.".

Definition 2 (Relation): In this chapter, we only consider the relation between two entities. An entity pair (E_i, E_j) represents the relation R_{ij} from entity E_i and E_j, where E_i is the first argument and E_j is the second argument. Relation R_{ij} take values that range over a set of relation class C^R. Note that (E_i, E_j) is an ordered pair, and there exists two relations $R_{ij} = (E_i, E_j)$ and $R_{ji} = (E_j, E_i)$ between entities E_i and E_j.

Definition 3 (Class): The class of an entity or relation is its decision attribute, which is one of the predefined class set and cannot be known before recognizing them. We denote the sets of predefined entity class and relation class as C^E and C^R respectively. All elements in C^E or C^R are mutually exclusive respectively.

Definition 4 (Constraint): A constraint is a 5-tuple $(r, \varepsilon^1, \varepsilon^2, \alpha_R, \alpha_\varepsilon)$. The symbols are defined as follows. $r \in C^R$ represents the class of a relation R_{ij}. $\varepsilon^1, \varepsilon^2 \in C^E$ represent the classes of the first argument E_i and the second argument E_j in the relation R_{ij} respectively. $\alpha_R \in [0,1]$ is real number that represents a joint conditional probability distribution of the classes of its two arguments, given the class r of the relation R_{ij}, i.e. $\alpha_R = \Pr\{\varepsilon^1, \varepsilon^2 \mid r\}$. $\alpha_\varepsilon \in [0,1]$ is real number that represents a conditional probability distribution of the class of a relation, given the classes of its two arguments ε^1 and ε^2, i.e. $\alpha_\varepsilon = \Pr\{r \mid \varepsilon^1, \varepsilon^2\}$. Note that α_R and α_ε can be easily learned from an annotated training dataset.

Definition 5 (Observation): We denote the observations of an entity and a relation in a sentence as O^E and O^R respectively. O^E or O^R represents all the "known" local properties of an entity or a relation, e.g., the spelling of a word, part-of-speech, and semantic related attributes acquired from external resources such as WordNet. The observations O^E and O^R can be viewed as random event, and $\Pr\{O^E\} = \Pr\{O^R\} \equiv 1$ because O^E and O^R in a given sentence are known.

Based on the above definitions, the question of entity and relation recognition can be described in a formal way as follows. Suppose in a given sentence, the set of entities is $\{E_1, E_2 \ldots E_n\}$, the set of relations is $\{R_{12}, R_{21}, R_{13}, R_{31}, \ldots, R_{1n}, R_{n1}, \ldots, R_{n-1,n}, R_{n,n-1}\}$, the predefined sets of entity classes and relation classes are $C^E = \{e_1, e_2, \ldots e_m\}$ and $C^R = \{r_1, r_2, \ldots r_k\}$ respectively, the observation of entity E_i is O_i^E, and the observation of relation R_{ij} is O_{ij}^R. n, m and k represent the number of entities, the number of the predefined entity classes and the number of the predefined relation classes respectively. The question is to search the most probable class assignment for each entity and relation of interest, given the observations of all entities and relations. In other words, the question is to solve the following two equations, using two kinds of constraint knowledge α_R and α_ε and the interaction among entities and relations illustrated as Figure 2.

$$e = \arg\max_d \Pr\{E_i = e_d \mid O_1^E, O_2^E, \cdots, O_n^E, O_{12}^R, O_{21}^R, \cdots, O_{1n}^R, O_{n1}^R, \cdots, O_{n-1,n}^R, O_{n,n-1}^R\} \quad (1)$$

$$r = \arg\max_d \Pr\{R_{ij} = r_d \mid O_1^E, O_2^E, \cdots, O_n^E, O_{12}^R, O_{21}^R, \cdots, O_{1n}^R, O_{n1}^R, \cdots, O_{n-1,n}^R, O_{n,n-1}^R\} \quad (2)$$

In equation (1), d =1, 2, …, m. And in equation (2), d=1, 2, …, k.

The Proposed Framework (Fan and Sun, 2005; Fan and Sun, 2006)

Because the class assignment of a single entity or relation depends not only on local attributes of itself, but also on those of all other entities and relations, the equation (1) and equation (2) cannot be solved directly. To simplify the problem, Fan and Sun presented the following method consisting of three stages:

At the first stage, the method employs two classifiers that perform subtasks entity recognition and relation recognition for each entity and each relation independently. Their outputs are used as conditional probability distributions $\Pr\{E|O^E\}$ and $\Pr\{R|O^R\}$ for each entity and relation, given the corresponding observations. Note that each of the local classifier could only depend on a large number of features, which are

not viewed as relevant to the latter inference process. The purpose at the first stage is to abstract away from the process of "inference with classifiers".

At the second stage, the class of each entity is determined by taking into account the classes of all the entities, as computed at the previous stage. This is achieved using the model dubbed "entity relation propagation diagram (**ERPD**)" and "entity relation propagation tree (**ERPT**)". During this stage, the constraint knowledge α_R in definition 4 and the conditional probability distribution $\Pr\{R|O^R\}$ for each relation, as computed at the first stage, are used.

At the third stage, each relation is recognized, while considering the classes of the involved entities produced during the second stage. During this stage, the constraint knowledge α_ε in definition 4 is used.

The procedure learning basic classifiers for entities and relations at the first stage is the same as Roth and YiH's approach (2002), which uses SnoW (Roth, 1998) as the prepositional learner. See, e.g., Roth and YiH (2002) for details. At the second stage, the aim of introducing ERPD is to estimate the conditional probability distribution $\Pr\{E|\,ERPD\}$ given the constraint α_R in definition 4 and the sets $\{\Pr\{E_i \mid O^{Ei}\}\}$ and $\{\Pr\{R_{ij} \mid O^{Rij}\}\}$ (i, j=1,…,n), as computed at the first stage.

For the readability, suppose $\Pr\{E|\,ERPD\}$ is given, the entity recognition equation (1) becomes the equation (3).

$$e = \begin{cases} \arg\max_j \Pr\{E_i = e_j \mid O_i^E\} & RV > \theta \\ \arg\max_j \Pr\{E_i = e_j \mid ERED\} & RV \le \theta \end{cases} \quad (3)$$

where, θ is a threshold determined experimentally. **RV** $\in$ [0, 1] is a real number, called the Reliable Value (**RV**), representing the belief degree of the output of the entity recognizer at the first stage. Suppose the maximum value of the conditional probability distribution of the target entity, given the observation, as Vm and the secondary maximum value as Vs, **RV** is defined as:

$$RV = \frac{V_m - V_s}{V_m + V_s} \quad (4)$$

The reason of introducing **RV** is due to a fact that only for ambiguous entities, it is effective by taking the classes of all entities in a sentence into account. **RV** measures whether an entity is ambiguous.

After we have recognized the classes of its two arguments in relation R, the relation can be recognized according to the following equation (5), i.e. the equation (2) becomes the following equation (5). The base idea is to search the probable relation among the different candidate relation classes given the observation, under satisfying the relation class constraint decided by the results of recognizing entity at the second stage.

$$r = \arg\max_k \Pr\{R = r^k \mid O_R\} \times W_R$$

$$W_R = \begin{cases} 1 & \text{if } \Pr\{r \mid \varepsilon^1, \varepsilon^2\} > 0 \\ 0 & \text{if } \Pr\{r \mid \varepsilon^1, \varepsilon^2\} = 0 \end{cases} \quad (5)$$

where, ε^1, ε^2 is the results of entity prediction at the second stage, $\Pr\{r|\ \varepsilon^1, \varepsilon^2\}$ is constraint knowledge α_ε in definition 4, and W_R is the weight of the constraint knowledge.

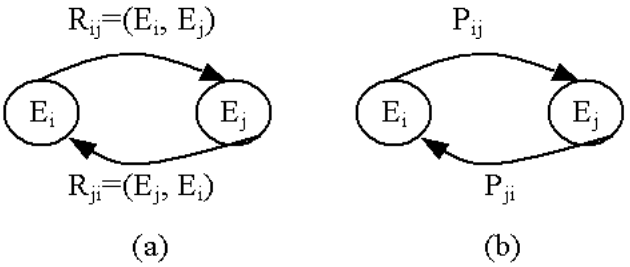
Entity Relation Propagation Diagram (Fan and Sun, 2005; Fan and Sun, 2006)

To fully represent the mutual dependencies, a model that can deal with cycles is designed for entity & relation recognition. Here, the model dubbed entity relation propagation diagram, which is similar to Causality Diagram (Fan, 2002; Fan et al., 2003) that has been applied on complex system fault diagnosis, is presented.

The classes of any two entities are dependent on each other through the relations, taking into account the relations between them. For example, the class of entity E_i in Figure 3(a) depends on the classes of relations R_{ij} and R_{ji} between entities E_i and E_j, and the classes of relations R_{ij} and R_{ji} depend on the classes of entities E_i and E_j. This means we can predict the class of the target entity according to the class of the neighboring entity, using the relations between them. Here, the relation reaction intensity is introduced to describe the prediction ability of this kind.

Definition 6 (Relation Reaction Intensity): We denote the relation reaction intensity from entity E_i to entity E_j as P_{ij}, which represents the ability that we can guess the class of E_j if we know the class of its neighboring entity E_i and the relation R_{ij} between them. The relation reaction intensity could be modeled using a condition probability

Figure 3. Illustration of the relation reaction between two entities



distribution $P_{ij}=Pr\{E_j|E_i\}$, and it is a matrix as follows.

$$\mathbf{P}_{ij}=\left[\mathbf{P}_{ij}^{1\bullet}\ \cdots\ \mathbf{P}_{ij}^{n\bullet}\right]=\begin{bmatrix} p_{ij}^{11} & \cdots & p_{ij}^{n1} \\ \vdots & \ddots & \\ p_{ij}^{n1} & \cdots & p_{ij}^{nn} \end{bmatrix}_{n\times n}$$

where n is the number of entity class. $\mathbf{P}_{ij}^{k\bullet}$ is the kth column of $\mathbf{P}_{ij}$, which represents the conditional probabilities of assigning different classes of entity E_j given the class of E_i as e_k.

According to the definition 3, all elements in C^E are mutually exclusive, thus we have

$$\sum_{l=1}^{n} p_{ij}^{kl}=1 \tag{6}$$

The element p_{ij}^{k} of P_{ij} represents the conditional probability $Pr\{E_j=e_l|E_i=e_k\}$, which is computed as follows.

$$\begin{aligned} p_{ij}^{kl} &= \Pr\{E_j=e_l \mid E_i=e_k\} \\ &= \sum_{t=1}^{N}\frac{\Pr\{R_{ij}=r_t\}\Pr\{E_i=e_k, E_j=e_l \mid R_{ij}=r_t\}}{\Pr\{E_i=e_k\}} \end{aligned}$$

According to definition 5, have

$$\Pr\{R_{ij}=r_t\}=\Pr\{R_{ij}=r_t \mid O_{ij}^{R}\},$$

$$\Pr\{E_i=e_k\}=\Pr\{E_i=e_k \mid O_i^{E}\}.$$

Then

$$p_{ij}^{kl}=\sum_{t=1}^{N}\frac{\Pr\{R_{ij}=r_t \mid O_{ij}^{R}\}\Pr\{E_i=e_k, E_j=e_l \mid R_{ij}=r_t\}}{\Pr\{E_i=e_k \mid O_i^{E}\}} \tag{7}$$

where $r_t \in C^R$, N is the number of relations in relation class set. In equation (7), $\Pr\{E_i=e_k, E_j=e_1 \mid R_{ij}=r_t\}$ represents the constraint knowledge α_R among entities and relations. $\Pr\{R_{ij}=r_t \mid O_{ij}^{R}\}$ and $\Pr\{E_i=e_k \mid O_i^{E}\}$ represent the outputs at the first stage.

Considering equation (7) together with equation (6), the result of computing according to equation (7) should be normalized according to equation (3). Thus, we have

$$\mathbf{P}_{ij}^{k\bullet}=\mathrm{Norm}\left(\begin{bmatrix} p_{ij}^{k1} \\ \vdots \\ p_{ij}^{kn} \end{bmatrix}\right) \tag{8}$$

where Norm(.) is a function which normalizes the vector in {}. For example,

$$\mathrm{Norm}\left(\begin{bmatrix} 0.5 \\ 0.7 \\ 0.8 \end{bmatrix}\right)=\begin{bmatrix} 0.5/(0.5+0.7+0.8) \\ 0.7/(0.5+0.7+0.8) \\ 0.8/(0.5+0.7+0.8) \end{bmatrix}=\begin{bmatrix} 0.25 \\ 0.35 \\ 0.4 \end{bmatrix}.$$

It is easy to see that $Pr\{E_i = e_k | O_i^E\}$ doesn't contribute to the normalized value. This means that the relation reaction intensity depends only on constraint knowledge α_R and the conditional probability distribution of the classes of relations given the observation.

Definition 7 (Observation Reaction Intensity): We denote the observation reaction intensity as the conditional probability distribution of an entity class, given the observation, i.e. $Pr\{E|O^E\}$, which is the output result at the first stage in the framework.

Entity Relation Propagation Diagram (ERPD) is a directed diagram, which allows cycles. As illustrated in Figure 4, the symbols used in the ERPD are defined as follows.

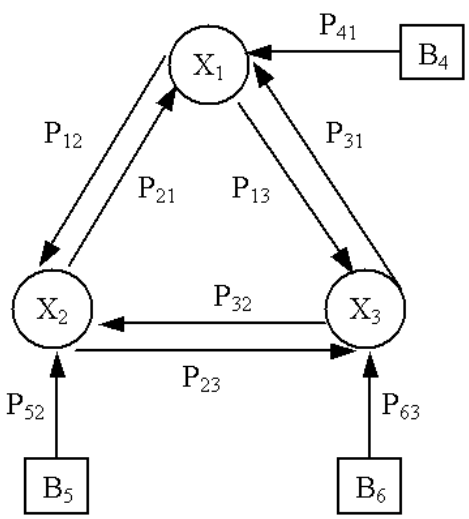
Circle: A node event variable that may equal any one of a set of mutually exclusive node events, which together cover the whole node sample space. Here, a node event variable represents an entity, a node event represents a predefined entity class, and the whole sample space represents the set of predefined entity classes. Box: A basic event is one of the independent sources of the linked node event variable. Here, a basic event represents the observation of an entity. Directed arc: A linkage event variable that may or may not enable an input event to cause the corresponding output event. Here, the linkage event variable from a node event variable to another node event variable represents relation reaction intensity in definition 6. And the linkage event variable from a basic event to the corresponding node event variable represents observation reaction intensity in definition 7. All arcs pointing to a node are logical OR relationship.

The method of computing the conditional probability distribution Pr{E| ERPD} based on the ERPD is described in the paper (Fan and Sun, 2006) for detail.

FUTURE TRENDS

The precision of recognizing entity affects directly the precision of recognizing relation between two entities. The future work will concentrate on how to improve the entity recognition. Based on the results in the paper (Fan and Sun, 2006), we think the possible solution is to define some additional relations between two entities. The additional relation is defined by experts and has the following characters: (1) it is easy to recognize by a relation classifier, i.e., its features are very rich and have strong discriminability. (2) This relation should

Figure 4. Illustration of an entity and relation propagation diagram



be helpful for the entity recognition. For example, the task is to recognize entity Clinton, White House and the relation between them. Suppose it is very difficult to recognize the entity Clinton and the relation between Clinton and White House, an additional relation spouse between Clinton and Hillary may be introduced because the relation spouse is easy to recognize and it is helpful to recognize entity Clinton.

CONCLUSION

Entity and relation recognition in text data is a key task in areas such as information extraction and question answering. The typically procedure for this question has several problems, (1) it cannot block errors propagation from entity recognizer to relation classifier, and (2) it cannot use some useful information that becomes available during relation recognition. The method based on the belief network has still several problems, (1) it cannot model the loopy dependencies among entities and relations, (2) it cannot use the constraint knowledge α_R, and (3) it cannot improve the entity recognition. The method based on entity relation propagation diagram (ERPD) has the following advantages, (1) entity relation propagation diagram can model the loopy dependencies among entities and relations, and (2) it can use two kinds of constraint knowledge α_R and α_ε learned from the annotated dataset. So, the method can improve not only relation recognition but also entity recognition in some degree.

ACKNOWLEDGMENTS

The research was supported by the National Natural Science Foundation of China under grant number 60703010, and the Natural Science Foundation of Chongqing province in China under grant number 2006BB2374.

REFERENCES

Changki Lee, Yi-Gyu Hwang, & Myung-Gil. (2007). Fine-Grained Named Entity Recognition and Relation Extraction for Question Answering. In Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. 799 – 800.

Califf, M., & Mooney, R. (1999). Relational Learning of Pattern-match Rules for Information Extraction. In Proceedings of the Sixteenth National Conference on Artificial Intelligence and Eleventh Conference on Innovative Applications of Artificial Intelligence. 328-334.

Chinchor, N. (1998). MUC-7 Information Extraction Task Definition. In Proceeding of the Seventh Message Understanding Conference (MUC-7), Appendices.

Freitag, D. (2000). Machine Learning for Information Extraction in Informal Domains. Machine learning. 39(2-3), 169-202.

Hirschman, L., Light, M., Breck, E., & Burger, J. (1999). Deep Read: A Reading Comprehension System. In Proceedings of the 37th Annual Meeting of Association for Computational Linguistics.

Pearl, J. (1998) Probability Reasoning in Intelligence Systems. Morgan Kaufmann.

Roth, D. (1998). Learning to resolve natural language ambiguities: A unified approach. In Proceeds of National Conference on Artificial Intelligence. 806-813.

Roth, D. and Yih, W. (2001). Relational Learning via Prepositional Algorithms: An Information Extraction Case Study. In Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence. 1257-1263.

Roth, D., & Yih, W. (2002). Probability Reasoning for Entity & Relation Recognition. In Proceedings of 20th International Conference on Computational Linguistics. 835-841.

Voorhees, E. (2000). Overview of the Trec-9 Question Answering Track. In The Ninth Text Retrieval Conference. 71-80.

Xinghua Fan. (2002). Causality Diagram Theory Research and Application on Fault Diagnosis of Complex System. Ph.D. Dissertation of Chongqing University, P.R. China.

Xinghua Fan, Qin Zhang, Maosong Sun, & Xiyue Huang. (2003). Reasoning Algorithm in Multi-Valued Causality Diagram. Chinese Journal of Computers. 26(3), 310-322.

Xinghua Fan, & Maosong Sun. (2005). A method of Recognizing Entity and Relation. In Proceedings of IJCNLP-2005, LNAI 3651. 245–256.

Xinghua Fan, & Maosong Sun. (2006). Knowledge Representation and Reasoning Based on Entity and Relation Propagation Diagram/Tree. Intelligent Data Analysis. 10(1), 81-102.

KEY TERMS

Entity and Relation Recognition (ERR): The entity and relation recognition is the task of assigning semantic classes (e.g., person, organization and location) to entities in a given sentence (i.e., Named Entity Recognition) and determining the relations (e.g., born-in and employee-of) that hold between the corresponding entities.

Entity Relation Propagation Diagram (ERPD): It is a directed diagram that allows cycles, in which an entity is considered as a node event variable and denoted as a circle, a node event represents a predefined entity class; the observation of an entity, i.e., the "known" local properties of an entity such as the spelling of a word, part-of-speech, is considered as a basic event and denoted as a box; the relation between two entities is transformed into a relation reaction intensity, is denoted as a directed arc pointed from the input event to the output event. An ERPD integrates all entities, their "known" local properties and the relation between two entities.

Information Extraction (**IE**): It is a sub-discipline of language engineering, a branch of computer science. It aims to apply methods and technologies from practical computer science to the problem of processing unstructured textual data automatically, with the objective to extract structured knowledge regarding a predefined domain.

Knowledge Representation (**KR**): Knowledge representation is the study of how knowledge about the world can be represented and what kinds of reasoning can be done with that knowledge.

Named Entity Recognition (NER): Named entities are atomic elements in text belonging to predefined categories such as the names of persons, organizations, locations, expressions of times, quantities, monetary values, percentages, etc. Named entity recognition is the task of identifying such named entities.

Relation Reaction Intensity: It is a condition probability distribution matrix, which represents the ability that we can guess the class of entity E_j if we know the class of its neighboring entity E_i and the relation R_{ij} between them.

Microarray Data Mining

Li-Min Fu
Southern California University of Health Sciences, USA

INTRODUCTION

Based on the concept of simultaneously studying the expression of a large number of genes, a DNA microarray is a chip on which numerous probes are placed for hybridization with a tissue sample. Biological complexity encoded by a deluge of microarray data is being translated into all sorts of computational, statistical or mathematical problems bearing on biological issues ranging from genetic control to signal transduction to metabolism. Microarray data mining is aimed to identify biologically significant genes and find patterns that reveal molecular network dynamics for reconstruction of genetic regulatory networks and pertinent metabolic pathways.

BACKGROUND

The idea of microarray-based assays seemed to emerge as early as of the 1980s (Ekins & Chu, 1999). In that period, a computer-based scanning and image-processing system was developed to quantify the expression level in tissue samples of each cloned complementary DNA sequence spotted in a 2D array on strips of nitrocellulose, which could be the first prototype of the DNA microarray. The microarray-based gene expression technology was actively pursued in the mid-1990s (Schena et al., 1998) and has seen rapid growth since then (Bier et al., 2008).

Microarray technology has catalyzed the development of the field known as functional genomics by offering high-throughput analysis of the functions of genes on a genomic scale (Schena et al., 1998). There are many important applications of this technology, including elucidation of the genetic basis for health and disease, discovery of biomarkers of therapeutic response, identification and validation of new molecular targets and modes of action, and so on. The accomplishment of decoding human genome sequence together with recent advances in the biochip technology has ushered in genomics-based medical therapeutics, diagnostics, and prognostics.

MAIN THRUST

The laboratory information management system (LIMS) keeps track of and manages data produced from each step in a microarray experiment, such as hybridization, scanning, and image processing. As microarray experiments generate a vast amount of data, the efficient storage and use of the data require a database management system. While some databases are designed to be data archives only, other databases such as ArrayDB (Ermolaeva et al., 1998) and Argus (Comander, Weber, Gimbrone, & Garcia-Cardena, 2001) allow information storage, query and retrieval as well as data processing, analysis and visualization. These databases also provide a means to link microarray data to other bioinformatics databases (e.g., NCBI Entrez systems, Unigene, KEGG, OMIM). The integration with external information is instrumental to the interpretation of patterns recognized in the gene-expression data. To facilitate the development of microarray databases and analysis tools, there is a need to establish a standard for recording and reporting microarray gene expression data. The MIAME (Minimum Information about Microarray Experiments) standard includes a description of experimental design, array design, samples, hybridization, measurements and normalization controls (Brazma et al., 2001).

Data Mining Objectives

Data mining addresses the question of how to discover a gold mine from historical or experimental data, particularly in a large database. The goal of data mining and knowledge discovery algorithms is to extract implicit, previously unknown and nontrivial patterns, regularities, or knowledge from large data sets that can be used to improve strategic planning and decision-mak-

ing. The discovered knowledge capturing the relations among the variables of interest can be formulated as a function for making prediction and classification or as a model for understanding the problem in a given domain. In the context of microarray data, the objectives are identifying significant genes and finding gene expression patterns associated with known or unknown categories. Microarray data mining is an important topic in bioinformatics, a field dealing with information processing on biological data, particularly, genomic data.

Practical Factors Prior to Data Mining

Some practical factors should be taken into account prior to microarray data mining. At first, microarray data produced by different platforms vary in their formats and may need to be processed differently. For example, one type of microarray with cDNA as probes produces ratio data from two channel outputs whereas another type of microarray using oligonucleotide probes generates non-ratio data from a single channel. Also, different platforms may pick up gene expression activity with different levels of sensitivity and specificity. Moreover, different data processing techniques may be required for different data formats.

Normalizing data to allow direct array-to-array comparison is a critical issue in array data analysis since there are several variables in microarray experiments that can affect measured mRNA levels (Schadt, Li, Ellis, & Wong, 2001; Yang et al., 2002). Variations may occur during sample handling, slide preparation, hybridization, or image analysis. Normalization is essential for correct microarray data interpretation. In simple ways, data can be normalized by dividing or subtracting expression values by a representative value (e.g., mean or median in an array) or by taking a linear transformation to zero mean and unit variance. As an example, data normalization in the case of cDNA arrays may proceed as follows. The local background intensity is first subtracted from the value of each spot on the array; and the two channels are normalized against the median values on that array; and then the Cy5/Cy3 fluorescence ratios and $\log_{10}$-transformed ratios are calculated from the normalized values. In addition, genes that do not change significantly can be removed through a filter in a process called data filtration.

While data analysis is a central issue in data mining, experimental design is critical as well. In particular, the use of replication in controlled experiments can significantly improve the outcome (Lee, Kuo, Whitmore, & Sklar, 2000).

Differential Gene Expression

To identify genes differentially expressed across two conditions is one of the most important issues in microarray data mining. In cancer research, for example, we wish to understand what genes are abnormally expressed in a certain type of cancer, so we conduct a microarray experiment and collect the gene expression profiles of normal and cancer tissues, respectively, as the control and test samples. The information regarding differential expression is derived from comparing the test against control sample.

To determine which genes are differentially expressed, a common approach is based on fold-change. In this approach, we simply decide a fold-change threshold (e.g., 2X) and select genes associated with changes greater than that threshold. If a cDNA microarray is used, the ratio of the test over control expression in a single array can be converted easily to fold change in both cases of up-regulation (induction) and down-regulation (suppression). For oligonucleotide chips, fold-change is computed from two arrays, one for test and the other for control sample. In this case, if multiple samples in each condition are available, the statistical t-test or Wilcoxon tests can be applied but the catch is that the Bonferroni adjustment to the level of significance on hypothesis testing would be necessary to account for the presence of multiple genes. The t-test determines the difference in mean expression values between two conditions and identifies genes with significant difference. The non-parametric Wilcoxon test is a good alternative in the case of non-Gaussian data distribution. SAM (Significance Analysis of Microarrays) (Tusher, Tibshirani, & Chu, 2001) is a state-of-the-art technique based on balanced perturbation of repeated measurements and minimization of the false discovery rate (FDR). FDR is the expected proportion of false positives among all declared positives. FDR, as an alternative to the *p*-value, has been widely accepted for gene selection from microarray data (Yang & Yang, 2006). In addition, multivariate statistical analysis techniques, such as singular value decomposition (Alter, Brown, & Botstein, 2000) and multi-dimensional scaling, can be applied to reduce the high dimensionality of microarray data.

Coordinated Gene Expression

Identifying genes that are co-expressed across multiple conditions is an issue with significant implications in microarray data mining. For example, given gene expression profiles measured over time, we are interested in knowing what genes are functionally related. The answer to this question also leads us to deduce the functions of unknown genes from their correlation with genes of known functions. Equally important is the problem of organizing samples based on their gene expression profiles so that distinct phenotypes or disease processes may be recognized or discovered.

The solutions to both problems are based on so-called cluster analysis, which is meant to group objects into clusters according to their similarity. For example, genes are clustered by their expression values across multiple conditions; samples are clustered by their expression values across genes. At issue is how to measure the similarity between objects. Two popular measures are the Euclidean distance and Pearson's correlation coefficient. Clustering algorithms can be divided into hierarchical and non-hierarchical (partitional). Hierarchical clustering is either agglomerative (starting with singletons and progressively merging) or divisive (starting with a single cluster and progressively breaking). Hierarchical agglomerative clustering is most commonly used in cluster analysis of microarray data. In this method, two most similar clusters are merged at each stage until all objects are included in a single cluster. The result is a dendrogram (a hierarchical tree) that encodes the relationships among objects by showing how clusters merge at each stage. Partitional clustering algorithms are best exemplified by *k*-means and self-organization maps (SOM). The fact that different clustering algorithms often yield dissimilar results has motivated an approach that seeks a consensus among them (Laderas & McWeeney, 2007).

Discriminant Analysis

Taking an action based on the category of the pattern recognized in microarray gene-expression data is an increasingly important approach to medical diagnosis and management (Furey et al., 2000; Golub et al., 1999; Khan et al., 2001). A class predictor derived on this basis can automatically discover the distinction between different classes of samples, independent of previous biological knowledge (Golub et al., 1999). Gene expression information appears to be a more reliable indicator than phenotypic information for categorizing the underlying causes of diseases. The microarray approach has offered hope for clinicians to arrive at more objective and accurate cancer diagnosis and hence choose more appropriate forms of treatment (Tibshirani, Hastie, Narasimhan, & Chu, 2002).

The central question is how to construct a reliable classifier that predicts the class of a sample on the basis of its gene expression profile. This is a pattern recognition problem and the type of analysis involved is referred to as discriminant analysis. In practice, given a limited number of samples, correct discriminant analysis must rely on the use of an effective gene selection technique to reduce the gene number and hence the data dimensionality. In general, genes that contribute most to classification as well as provide biological insight are selected. Approaches to discriminant analysis range from statistical analysis (Golub et al., 1999) and a Bayesian model (Lee, Sha, Dougherty, Vannucci, & Mallick, 2003) to Fisher's linear discriminant analysis (Xiong, Li, Zhao, Jin, & Boerwinkle, 2001) and support vector machines (SVMs) (Guyon, Weston, Barnhill, & Vapnik, 2002).

Gene Annotation and Pathway Analysis

Differential and coordinated gene expression analyses result in genes or gene clusters critical to the issue under investigation. These results can be further annotated using existing knowledge concerning gene functional classes and gene networks. To this end, an important bioinformatics source is the Gene Ontology (GO) database (at http://www.geneontology.org). This database provides information in the aspects of biological processes, cellular components, and molecular functions for genes under query. Such information can enhance microarray analysis and may also provide cognitive advantages (Lee, Katari, & Sachidanandam, 2005; Liu, Hughes-Oliver, & Menius, 2007).

To reconstruct biological pathways from gene expression data represents a great challenge for microarray data mining. This problem can be tackled more efficiently by capitalizing on existing knowledge. KEGG (the Kyoto Encyclopedia of Genes and Genomics) records networks of molecular interactions on any organism and is instrumental to analysis of the functions of genes and their interactions (Kanehisa & Goto, 2000). Engineering and statistical approaches to

the problem have been developed, notably, the Bayesian network approach. Bayesian networks are graph-based models of joint multivariate probability distributions capturing conditional dependency and independency among variables. The Bayesian approach has been applied to construct gene regulatory networks from gene expression data with success (Friedman, Linial, Nachman, & Pe'er, 2000; Polanski, Polanska, Jarzab, Wiench, & Jarzab, 2007).

Microarray Data Mining Applications

The microarray technology permits a large-scale analysis of gene functions in a genomic perspective and has brought about important changes in how we conduct basic research in science and practice in clinical medicine. There have existed an increasing number of applications with this technology. Table 1 summarizes some of most important microarray data mining problems and their solutions.

Only the minority of all the yeast (Saccharomyces cerevisiae) open reading frames in the genome sequence could be functionally annotated on the basis of sequence information alone (Zweiger, 1999), while microarray results showed that nearly 90% of all yeast mRNAs (messenger RNAs) are observed to be present (Wodicka, Dong, Mittmann, Ho, & Lockhart, 1997). Functional annotation of a newly discovered gene based on sequence comparison with other known gene sequences is sometimes misleading. Microarray-based genome-wide gene expression analysis has made it possible to deduce the functions of novel or poorly characterized genes from co-expression with already known genes (Eisen, Spellman, Brown, & Botstein, 1998). The microarray technology is a valuable tool for measuring whole-genome mRNA and enables system-level exploration of transcriptional regulatory networks (Cho et al., 1998; DeRisi, Iyer, & Brown, 1997; Laub, McAdams, Feldblyum, Fraser, & Shapiro, 2000; Tavazoie, Hughes, Campbell, Cho, & Church, 1999). Hierarchical clustering can help us recognize genes whose cis-regulatory elements are bound by the same proteins (transcription factors) in vivo. Such a set of co-regulated genes is known as a "regulon". Statistical characterization of known regulons is used to derive criteria for inferring new regulatory elements. To identify regulatory elements and associated transcription factors is fundamental to building a global gene regulatory network essential for understanding the genetic control and biology in living cells.

The limitation of the morphology-based approach to cancer classification has led to molecular classifica-

Table 1. Three common computational problems in microarray data mining

Problem I: To identify differentially expressed genes, given microarray gene expression data collected in two conditions, types, or states.

[Solutions:]
- Fold change
- t-test or Wilcoxon rank sum test (with Bonferroni's correction)
- Significance analysis of microarrays

Problem II: To identify genes expressed in a coordinated manner, given microarray gene expression data collected across a set of conditions or time points.

[Solutions:]
- Hierarchical clustering
- Self-organization
- *k*-means clustering

Problem III: To select genes for discriminant analysis, given microarray gene expression data of two or more classes.

[Solutions:]
- Neighborhood analysis
- Support vector machines
- Principal component analysis
- Bayesian analysis
- Fisher's linear discriminant analysis

tion. Techniques such as immunohistochemistry and RT-PCR are used to detect cancer-specific molecular markers, but pathognomonic molecular markers are unfortunately not available for most of solid tumors (Ramaswamy et al., 2001). Furthermore, molecular markers do not guarantee a definitive diagnosis owing to possible failure of detection or presence of marker variants. The approach of constructing a classifier based on gene expression profiles has gained increasing interest, following the success in demonstrating that microarray data differentiated between two types of leukemia (Golub et al., 1999). In this application, the two data-mining problems are to identify gene expression patterns or signatures associated with each type of leukemia and to discover subtypes within each. The first problem is dealt with by gene selection, and the second one by cluster analysis. Table 2 lists some application examples of microarray data mining.

FUTURE TRENDS

The future challenge is to realize biological networks that provide qualitative and quantitative understanding of molecular logic and dynamics. To meet this challenge, recent research has begun to focus on leveraging prior biological knowledge and integration with biological analysis in quest of biological truth. The integration of bioinformatics and system biology has carried microarray data mining to the next level.

CONCLUSION

Microarray technology has rapidly emerged as a powerful tool for biological research and clinical investigation. However, the large quantity and complex nature of data produced in microarray experiments often plague researchers who are interested in using this technology. Microarray data mining uses specific data processing and normalization strategies and has its own objectives, and it requires effective computational algorithms and statistical techniques to arrive at valid results. The microarray technology has been perceived as a revolutionary technology in biomedicine, but the hardware device does not pay off unless backed up with sound data mining software.

REFERENCES

Alter, O., Brown, P. O., & Botstein, D. (2000). Singular value decomposition for genome-wide expression data processing and modeling. *Proc Natl Acad Sci U S A, 97*(18), 10101-10106.

Table 2. Examples of microarray data mining applications

Classical Work:

- Identified functional related genes and their genetic control upon metabolic shift from fermentation to respiration (DeRisi et al., 1997).
- Explored co-expressed or co-regulated gene families by cluster analysis (Eisen et al., 1998).
- Determined genetic network architecture based on coordinated gene expression analysis and promoter motif analysis (Tavazoie et al., 1999).
- Differentiated acute myeloid leukemia from acute lymphoblastic leukemia by selecting genes and constructing a classifier for discriminant analysis (Golub et al., 1999).
- Selected genes differentially expressed in response to ionizing radiation based on significance analysis (Tusher et al., 2001).

Recent Work:

- Analyzed gene expression in the Arabidopsis genome (Yamada et al., 2003).
- Discovered conserved genetic modules (Stuart, Segal, Koller, & Kim, 2003).
- Elucidated functional properties of genetic networks and identified regulatory genes and their target genes (Gardner, di Bernardo, Lorenz, & Collins, 2003).
- Identified genes associated with Alzheimer's disease (Roy Walker et al., 2004).
- Detected viruses in human tumors (Li et al., 2006).
- Identified chemosensitivity-related genes of breast cancer (Ikeda, Jinno, & Shirane, 2007).

Bier, F. F., von Nickisch-Rosenegk, M., Ehrentreich-Forster, E., et al. (2008). DNA Microarrays. *Adv Biochem Eng Biotechnol, 109*, 433-453.

Brazma, A., Hingamp, P., Quackenbush, J., et al. (2001). Minimum information about a microarray experiment (MIAME)-toward standards for microarray data. *Nat Genet, 29*(4), 365-371.

Cho, R. J., Campbell, M. J., Winzeler, E. A., et al. (1998). A genome-wide transcriptional analysis of the mitotic cell cycle. *Mol Cell, 2*(1), 65-73.

Comander, J., Weber, G. M., Gimbrone, M. A., Jr., et al. (2001). Argus--a new database system for Web-based analysis of multiple microarray data sets. *Genome Res, 11*(9), 1603-1610.

DeRisi, J. L., Iyer, V. R., & Brown, P. O. (1997). Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science, 278*(5338), 680-686.

Eisen, M. B., Spellman, P. T., Brown, P. O., et al. (1998). Cluster analysis and display of genome-wide expression patterns. *Proc Natl Acad Sci U S A, 95*(25), 14863-14868.

Ekins, R., & Chu, F. W. (1999). Microarrays: their origins and applications. *Trends Biotechnol, 17*(6), 217-218.

Ermolaeva, O., Rastogi, M., Pruitt, K. D., et al. (1998). Data management and analysis for gene expression arrays. *Nat Genet, 20*(1), 19-23.

Friedman, N., Linial, M., Nachman, I., et al. (2000). Using Bayesian networks to analyze expression data. *J Comput Biol, 7*(3-4), 601-620.

Furey, T. S., Cristianini, N., Duffy, N., et al. (2000). Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics, 16*(10), 906-914.

Gardner, T. S., di Bernardo, D., Lorenz, D., et al. (2003). Inferring genetic networks and identifying compound mode of action via expression profiling. *Science, 301*(5629), 102-105.

Golub, T. R., Slonim, D. K., Tamayo, P., et al. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science, 286*(5439), 531-537.

Guyon, I., Weston, J., Barnhill, S., et al. (2002). Gene selection for cancer classification using support vector machines. *machine learning, 46*(1/3), 389-422.

Ikeda, T., Jinno, H., & Shirane, M. (2007). Chemosensitivity-related genes of breast cancer detected by DNA microarray. *Anticancer Res, 27*(4C), 2649-2655.

Kanehisa, M., & Goto, S. (2000). KEGG: kyoto encyclopedia of genes and genomes. *Nucleic Acids Res, 28*(1), 27-30.

Khan, J., Wei, J. S., Ringner, M., et al. (2001). Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nat Med, 7*(6), 673-679.

Laderas, T., & McWeeney, S. (2007). Consensus framework for exploring microarray data using multiple clustering methods. *Omics, 11*(1), 116-128.

Laub, M. T., McAdams, H. H., Feldblyum, T., et al. (2000). Global analysis of the genetic network controlling a bacterial cell cycle. *Science, 290*(5499), 2144-2148.

Lee, J. S., Katari, G., & Sachidanandam, R. (2005). GObar: a gene ontology based analysis and visualization tool for gene sets. *BMC Bioinformatics, 6*, 189.

Lee, K. E., Sha, N., Dougherty, E. R., et al. (2003). Gene selection: a Bayesian variable selection approach. *Bioinformatics, 19*(1), 90-97.

Lee, M. L., Kuo, F. C., Whitmore, G. A., et al. (2000). Importance of replication in microarray gene expression studies: statistical methods and evidence from repetitive cDNA hybridizations. *Proc Natl Acad Sci U S A, 97*(18), 9834-9839.

Li, C., Chen, R. S., Hung, S. K., et al. (2006). Detection of Epstein-Barr virus infection and gene expression in human tumors by microarray analysis. *J Virol Methods, 133*(2), 158-166.

Liu, J., Hughes-Oliver, J. M., & Menius, J. A., Jr. (2007). Domain-enhanced analysis of microarray data using GO annotations. *Bioinformatics, 23*(10), 1225-1234.

Polanski, A., Polanska, J., Jarzab, M., et al. (2007). Application of Bayesian networks for inferring cause-effect relations from gene expression profiles of cancer versus normal cells. *Math Biosci, 209*(2), 528-546.

Ramaswamy, S., Tamayo, P., Rifkin, R., et al. (2001). Multiclass cancer diagnosis using tumor gene expression signatures. *Proc Natl Acad Sci U S A, 98*(26), 15149-15154.

Roy Walker, P., Smith, B., Liu, Q. Y., et al. (2004). Data mining of gene expression changes in Alzheimer brain. *Artif Intell Med, 31*(2), 137-154.

Schadt, E. E., Li, C., Ellis, B., et al. (2001). Feature extraction and normalization algorithms for high-density oligonucleotide gene expression array data. *J Cell Biochem, Suppl 37*, 120-125.

Schena, M., Heller, R. A., Theriault, T. P., et al. (1998). Microarrays: biotechnology's discovery platform for functional genomics. *Trends Biotechnol, 16*(7), 301-306.

Stuart, J. M., Segal, E., Koller, D., et al. (2003). A gene-coexpression network for global discovery of conserved genetic modules. *Science, 302*(5643), 249-255.

Tavazoie, S., Hughes, J. D., Campbell, M. J., et al. (1999). Systematic determination of genetic network architecture. *Nat Genet, 22*(3), 281-285.

Tibshirani, R., Hastie, T., Narasimhan, B., et al. (2002). Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proc Natl Acad Sci U S A, 99*(10), 6567-6572.

Tusher, V. G., Tibshirani, R., & Chu, G. (2001). Significance analysis of microarrays applied to the ionizing radiation response. *Proc Natl Acad Sci U S A, 98*(9), 5116-5121.

Wodicka, L., Dong, H., Mittmann, M., et al. (1997). Genome-wide expression monitoring in Saccharomyces cerevisiae. *Nat Biotechnol, 15*(13), 1359-1367.

Xiong, M., Li, W., Zhao, J., et al. (2001). Feature (gene) selection in gene expression-based tumor classification. *Mol Genet Metab, 73*(3), 239-247.

Yamada, K., Lim, J., Dale, J. M., et al. (2003). Empirical analysis of transcriptional activity in the Arabidopsis genome. *Science, 302*(5646), 842-846.

Yang, J. J., & Yang, M. C. (2006). An improved procedure for gene selection from microarray experiments using false discovery rate criterion. *BMC Bioinformatics, 7*, 15.

Yang, Y. H., Dudoit, S., Luu, P., et al. (2002). Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Res, 30*(4), e15.

Zweiger, G. (1999). Knowledge discovery in gene-expression-microarray data: mining the information output of the genome. *Trends Biotechnol, 17*(11), 429-436.

KEY TERMS

Bioinformatics: All aspects of information processing on biological data, in particular, genomic data. The rise of bioinformatics is driven by the genomic projects.

cis-Regulatory Element: The genetic region that affects the activity of a gene on the same DNA molecule.

Clustering: The process of grouping objects according to their similarity. This is an important approach to microarray data mining.

Functional Genomics: The study of gene functions on a genomic scale, especially, based on microarrays.

Gene Expression: Production of mRNA from DNA (a process known as transcription) and production of protein from mRNA (a process known as translation). Microarrays are used to measure the level of gene expression in a tissue or cell.

Gene Ontology: A controlled vocabulary used to describe the biology of genes and gene products in any organism.

Genomic Medicine: Integration of genomic and clinical data for medical decision.

Microarray: A chip on which numerous probes are placed for hybridization with a tissue sample to analyze its gene expression.

Post-Genome Era: The time after the complete human genome sequence is decoded.

Transcription Factor: A protein that binds to the cis-element of a gene and affects its expression.

Minimum Description Length Adaptive Bayesian Mining

Diego Liberati
Italian National Research Council, Italy

INTRODUCTION

In everyday life, it often turns out that one has to face a huge amount of data, often not completely homogeneous, often without an immediate grasp of an underlying simple structure. Many records, each instantiating many variables are usually collected with the help of several tools.

Given the opportunity to have so many records on several possible variables, one of the typical goals one has in mind is to classify subjects on the basis of a hopefully reduced meaningful subset of the measured variables.

The complexity of the problem makes it natural to resort to automatic classification procedures (Duda and Hart, 1973) (Hand et al., 2001). Then, a further questions could arise, like trying to infer a synthetic mathematical and/or logical model, able to capture the most important relations between the most influencing variables, while pruning (O'Connel 1974) the not relevant ones. Such interrelated aspects will be the focus of the present contribution.

In the First Edition of this encyclopedia we already introduced three techniques dealing with such problems in a pair of articles (Liberati, 2005) (Liberati et al., 2005). Their rationale is briefly recalled in the following background section in order to introduce the kind of problems also faced by the different approach described in the present article, which will instead resort to the Adaptive Bayesian Networks implemented by Yarmus (2003) on a commercial wide spread data base tool like Oracle.

Focus of the present article will thus be the use of Adaptive Bayesian Networks are in order to unsupervisedly learn a classifier direcly form data, whose minimal set of features is derived through the classical Minimun Description Lenght (Barron and Rissanen, 1998) popular in information theory.

Reference will be again made to the same popular micro-arrays data set also used in (Liberati et al., 2005), not just to have a common benchmark useful to compare results and discuss complementary advantages of the various procedures, but also because of the increasing relevance of the bioinformatics field itself.

BACKGROUND

The introduced tasks of selecting salient variables, identifying their relationships from data and infer a logical and/or dynamical model of interaction may be sequentially accomplished with various degrees of success in a variety of ways.

In order to reduce the dimensionality of the problem, thus simplifying both the computation and the subsequent understanding of the solution, the critical problem of selecting the most relevant variables must be solved.

The very simple approach to resort to cascading a Divisive Partitioning of data orthogonal to the Principal Directions – PDDP - (Boley 1998) and *k-means*, already proven to be a good way to initialize k-means (Savaresi and Booley, 2004) and to be successful in the context of analyzing the logs of an important telecom provider (Garatti et al., 2004), was presented in (Liberati et al., 2005) with reference to a paradigmatic case of micro-arrays data in bioinformatics

A more sophisticated possible approach is to resort to a rule induction method, like the one described as Hamming Clustering in Muselli and Liberati (2000). Such a strategy also offers the advantage to extract underlying rules, implying conjunctions and/or disjunctions between the identified salient variables. Thus, a first idea of their even non-linear relations is provided as a first step to design a representative model, whose variables will be the selected ones. Such an approach has been shown (Muselli and Liberati, 2002) to be not less powerful over several benchmarks than the popular decision tree developed by Quinlan (1994). Then, a possible approach to blindly build a simple linear approximating model is to resort to piece-wise affine (PWA) identification of hybrid systems (Ferrari-Trecate

et al., 2003). The cascading of such two last approaches has been proposed in (Liberati, 2005).

Here just a few more approaches are recalled among the most popular ones, to whom the ones used either here or in either (Liberati, 2005) or (Liberati et al., 2005) are in some way comparable. For a widest bibliography, one could refer to both edition of this Encyclopedia, and in particular to the bibliography cited in the referenced papers. Among the simplest approaches, principal components (MacQueen, 1967) (Golub and van Loan, 1996) help to order the variables from the most relevant to the least one, but only under a linear possibly homogeneous framework. Partial least squares do allow to extend to non-linear models, provided that one has prior information on the structure of the involved non-linearity; in fact, the regression equation needs to be written before identifying its parameters. Clustering (Kaufman and Rousseeuw, 1990) (Jain and Dubes, 1998) (Jain at al., 1999) is instead able to operate even in an unsupervised way, without the a priori correct classification of a training set, but even fuzzy (Karayiannis and Bezdek 1997) and supervised (Setnes, 2000) approaches have been explored. Neural networks are known to learn the embedded rules, but their possibility to make rules explicit (Taha & Ghosh, 1999) or to underline the salient variables is only indirect. Support Vector Machines (Vapnik, 1998) are a very simple and popular general purpose approach whose theoretic foundation makes it worth in many applications.

MAIN THRUST OF THE CHAPTER

Adaptive Bayesian Networks

A learning strategy searching for a trade-off between a high predictive accuracy of the classifier and a low cardinality of the selected feature subset may be derived according to the central hypothesis that a good feature subset should contain features that are highly correlated with the class to be predicted, yet uncorrelated with each other.

Based on information theory, the Minimum Description Length (MDL) principle (Barron and Rissanen, 1998) provides the statement that the best theory to infer from training data is the one that minimizes both the length (i.e. the complexity) of the theory itself and the length of the data encoded with respect to it. In particular, MDL can thus be employed as a criteria to judge the quality of a classification model.

The motivation underlying the MDL method is to find a compact encoding of the training data. To this end, the MDL measure introduced in Friedman et al. (1997) can be adopted, weighting how many bits one do need to encode the specific model (i.e. its length), and how many bits are needed to describe the data based on the probability distribution associated to the model.

This approach can be applied to address the problem of feature selection, by considering each feature alone as a simple predictive model of the target class. As described in (Kononenko1995), each feature can be ranked according to its description length, that reflects the strength of its correlation with the target. In this context, the MDL measure is given by Yarmus (2003), again weighting the encoding length, where one have one sub-model for each value of the feature, with the number of bits needed to describe the data, based on the probability distribution of the target value associated to each sub-model.

However, once all features have been ordered by rank, no a priori criterion is available to choose the cut-off point beyond which features can be discarded. To circumvent this drawback, one can start with building a classifier on the set of the n-top ranked features. Then, a new feature is sequentially added to this set, and a new classifier is built, until no improvement in accuracy is achieved.

Our approach takes into account two different classifiers derived from Bayesian Networks, i.e. the Naïve Bayes (NB) and the Adaptive Bayesian Network (ABN).

NB is a very simple Bayesian network consisting of a special node (i.e. the target class) that is parent of all other nodes (i.e. the features or attributes) that are assumed to be conditionally independent, given the value of the class. The NB network can be "quantified" against a training dataset of pre-classified instances, i.e. we can compute the probability associated to a specific value of each attribute, given the value of the class label. Then, any new instance can be easily classified making use of the Bayes rule. Despite its strong independence assumption is clearly unrealistic in several application domains, NB has been shown to be competitive with more complex state-of-the-art classifiers (Friedman et al., 1997) (Keogh and Pazzani., 2002) (Cheng and Greiner,1999).

In the last years, a lot of research has focused on improving NB classifiers by relaxing their full independence assumption. One of the most interesting approaches is based on the idea of adding correlation arcs between the attributes of a NB classifier.

Specific structural constraints are imposed on these "augmenting arcs" (Friedman et al., 1997) (Keogh and Pazzani., 2002), in order to maintain computational simplicity on learning. The algorithm here proposed, the Adaptive Bayesian Network (ABN) (Yarmus, 2003), is a greedy variant, based on MDL, of the approach proposed in (Keogh and Pazzani., 2002).

In brief, the steps needed to build an ABN classifier are the following. First, the attributes (predictors) are ranked according to their MDL importance. Then, the network is initialized to NB on the top k ranked predictors, that are treated as conditionally independent. Next, the algorithm attempts to extend NB by constructing a set of tree-like multi-dimensional features.

Feature construction proceeds as follows. The top ranked predictor is stated as a seed feature, and the predictor that most improves feature predictive accuracy, if any, is added to the seed. Further predictors are added in such a way to form a tree structure, until the accuracy does not improve.

Using the next available top ranked predictor as a seed, the algorithm attempts to construct additional features in the same manner. The process is interrupted when the overall predictive accuracy cannot be further improved or after some pre-selected number of steps.

The resulting network structure consists of a set of conditionally independent multi-attribute features, and the target class probabilities are estimated by the product of feature probabilities. Interestingly, each multi-dimensional feature can be expressed in terms of a set of if-then rules enabling users to easily understand the basis of model predictions.

A Paradigmatic Example: *The Leukemia Classification*

Data are taken from a public repository often adopted as a reference benchmark (Golub et al., 1999) in order to test new classification techniques and compare and complement it with the other available methodologies. Such database is constituted by gene expression data over 72 subjects, relying on 7,129 genes. Of the 72 subjects, 47 are cases of acute lymphoblastic leukaemia (ALL), while the remaining 25 are cases of acute myeloid leukaemia (AML).

An experimental bottleneck in this kind of experiment, like often in several applications, is the difficulty in collecting a high number of homogeneous subjects in each class of interest, making the classification problem even harder: not only a big matrix is involved, but such matrix has a huge number of variables (7,129 genes) with only a very poor number of samples (72 subjects).

On such data set, two series of experiments for learning both NB and ABN classifiers have been supervisedly performed on the 38 samples of the training set already defined in (Golub et al., 1999) in order to be able to compare results. We started with the set of the first 10 predictors found by MDL, and added the following attributes, one by one, feeding them to both NB and ABN classifiers, until no further improvement in accuracy was achieved.

Their predictive accuracy has been evaluated by the rate of correct predictions on the 34 samples of the test set also defined in (Golub et al., 1999). It is worth noticing that the classifiers built with 15 and 20 predictors result in the same accuracy. Interestingly, the only errors of both classifiers consist again in the misclassification of AML samples (i.e. 3 samples for NB and 2 of them, same as in (Golub et al 1999), for ABN).

Interestingly, 12 of the 15 more salient genes found are among the 25 genes most highly correlated with AML in (Golub et al., 1999). None of them is instead among the 25 most highly correlated with ALL in (Golubet al., 1999).

Not surprinsingly, the 2 more salient genes identified via the PDDP + *k-means* approach described in (Liberati et al., 2005) are among the top 15 of the BN approach: even more, the more salient there is the 15th here, after which no improvement is obtained by adding new genes. Finally, the only error performed among the subjects in the test set used here and in (Golub at., 1999) by the unsupervised approach proposed in (Liberati et al., 2005) is one of the 2 errors of ABN

FUTURE TRENDS

Further work is under evaluation in order to infer rules and identify models from this same example with various approaches, as well as approaching different prob-

lems with several recalled approaches, most of whom originally developed each one for different purposes and tested over different benchmarks.

The joint use of the approaches briefly described in the present contribution, together with (some of) those recalled, starting from data without known priors about their relationships, will hopefully allows to reduce dimensionality without significant loss in information, then to infer logical relationships, and, finally, to identify a simple input-output model of the involved process that also could be used for controlling purposes in the systems biology approach to investigate cell dynamics.

The definition (Bosin et al., 2006) of a collaborative framework for e-Science sharing across physical locations could also become a powerful distributed ICT tool for sharing delocalized expertise and data also in data mining approaches to complex problems

CONCLUSION

The proposed approach seems to be consistent at least with the simple clustering (Liberati et al., 2005) also proposed in the first edition of this same Encyclopedia and with the more articulated procedure originally proposed by (Golub et al. 1999), at least on the particular application described, which is anyway representative of quite a general class of not so easy problems even beyond bioinformatics: it can thus represent one more powerful tool for quite a wide spectrum of applications in, and beyond, data mining, providing an up-to-date answer to the quest of formally extracting knowledge from data and helping to sketch a model of the underlying process.

The fact that a combination of different approaches, taken from partially complementary disciplines, proves to be effective may indicate a fruitful direction in combining in different ways classical and new approaches to improve classification even in critical fields.

REFERENCES

Barron A., Rissanen J., Yu B., (1998) The minimum description length principle in coding and modelling, IEEE Transactions on Information Theory, 44: 2743-2760.

Boley, D.L. (1998). Principal direction divisive partitioning, Data Mining and Knowledge Discovery, 2(4), 325-344.

Cheng G., Greiner R., (1999). Comparing Bayesian Network Classifiers, Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers, Inc., San Francisco.

Duda, R.O., & Hart, P.E. (1973). Pattern classification and scene analysis. New York: Wiley.

Ferrari-Trecate, G., Muselli, M., Liberati, D., Morari, M. (2003). A clustering technique for the identification of piecewise affine systems. Automatica, 39, 205-217.

Friedman N., Geiger D., Goldszmidt M. (1997). Bayesian Network Classifiers, Machine Learning, 29: 131-161.

Garatti, S., Savaresi, S., & Bittanti, S. (2004). On the relationships between user profiles and navigation sessions in virtual communities: A data-mining approach,. Intelligent Data Analysis.

Golub, G.H., & van Loan, C.F. (1996). Matrix computations. Johns Hopkins University Press.

Golub, T.R., et al. (1999). Molecular classification of cancer: Class discovery and class prediction by gene expression. *Science, 286*, 531-537.

Hand, D., Mannila, H., & Smyth, P. (2001). Principles of data-mining. Cambridge, MA: MIT Press.

Jain, A., & Dubes, R. (1998). Algorithms for clustering data. London: Sage Publications.

Jain, A.K., Murty, M.N., & Flynn, P.J. (1999). Data clustering: A review. ACM Computing Surveys, 31, 264-323.

Karayiannis, N.B,. & Bezdek, J.C. (1997). An integrated approach to fuzzy learning vector quantization and fuzzy C-means clustering. IEEE Trans. Fuzzy Systems, 5, 622-628.

Kaufman, L., & Rousseeuw, P.J. (1990). Finding groups in data: An introduction to cluster analysis. New York: Wiley.

Keogh E., Pazzani M.J. (2002), Learning the structure of augmented Bayesian classifiers, International Journal on Artificial Intelligence Tools, Vol. 11, No. 4, 587-601

.Kononenko I. (1995), On biases in estimating multi-valued attributes, IJCAI95, 1034-1040.

Liberati D (2005) Model identification through data mining, Encyclopedia of data warehousing and mining: 820-825, J. Wang ed, Idea Book, Hershey, PA, 2005

Liberati D, Garatti S., Bittanti S (2005) Unsupervised mining of genes classifying leukemia, Encyclopedia of data warehousing and mining: 1155-1159, J. Wang ed, Idea Book, Hershey, PA

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. Proceeedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, California.

Muselli, M., & Liberati, D. (2000). Training digital circuits with Hamming clustering. IEEE Transactions on Circuits and Systems—I: Fundamental Theory and Applications, 47, 513-527.

Muselli, M., & Liberati, D. (2002). Binary rule generation via Hamming clustering. IEEE Transactions on Knowledge and Data Engineering, 14, 1258-1268.

O'Connel, M.J. (1974). Search program for significant variables. Comp. Phys. Comm., 8, 49.

Quinlan, J.R. (1994). C4.5: Programs for machine learning. San Francisco: Morgan Kaufmann.

Savaresi, S.M., & Boley, D.L. (2004). A comparative analysis on the bisecting K-means and the PDDP clustering algorithms. International Journal on Intelligent Data Analysis,8(4): 345-363

Setnes, M. (2000). Supervised fuzzy clustering for rule extraction. IEEE Trans. Fuzzy Systems, 8, 416-424.

Taha, I., & Ghosh, J. (1999). Symbolic interpretation of artificial neural networks. IEEE T Knowledge and Data Eng., 11, 448-463.

Vapnik, V. (1998). Statistical learning theory. New York: Wiley.

Yarmus J.S. (2003), ABN: A Fast, Greedy Bayesian Network Classifier,. http://otn.oracle.com/products/bi/pdf/adaptive_bayes_net.pdf

KEY TERMS

Hamming Clustering: A fast binary rule generator and variable selector able to build understandable logical expressions by analyzing the Hamming distance between samples.

Hybrid Systems: Their evolution in time suddenly switches from a smooth dynamics to a different one.

***k-means*:** Iterative clustering technique subdividing the data in such a way to maximize the distance among centroids of different clusters, while minimizing the distance among data within each cluster. It is sensitive to initialization.

Identification: Definition of the structure and computation of its parameters best suited to mathematically describe the process underlying the data.

PDDP (Principal Direction Divisive Partitioning): One-shot clustering technique based on principal component analysis and singular value decomposition of the data, thus partitioning the dataset according to the direction of maximum variance of the data. It is used here in order to initialize K-means.

Principal Component Analysis: Rearrangement of the data matrix in new orthogonal transformed variables ordered in decreasing order of variance.

Rule Inference: The extraction from the data of the embedded synthetic logical description of their relationships.

Relevant (Salient) Variable*s:* The real important ones among the many apparently involved in a process.

Unsupervised Clustering: Automatic classification of a dataset in two of more subsets on the basis of the intrinsic properties of the data without taking into account further contextual information.

Mining 3D Shape Data for Morphometric Pattern Discovery

Li Shen
University of Massachusetts Dartmouth, USA

Fillia Makedon
University of Texas at Arlington, USA

INTRODUCTION

Recent technological advances in 3D digitizing, non-invasive scanning, and interactive authoring have resulted in an explosive growth of 3D models in the digital world. There is a critical need to develop new 3D data mining techniques for facilitating the indexing, retrieval, clustering, comparison, and analysis of large collections of 3D models. These approaches will have important impacts in numerous applications including multimedia databases and mining, industrial design, biomedical imaging, bioinformatics, computer vision, and graphics.

For example, in similarity search, new shape indexing schemes (e.g. (Funkhouser et al., 2003)) are studied for retrieving similar objects from databases of 3D models. These shape indices are designed to be quick to compute, concise to store, and easy to index, and so they are often relatively compact. In computer vision and medical imaging, more powerful shape descriptors are developed for morphometric pattern discovery (e.g., (Bookstein, 1997; Cootes, Taylor, Cooper, & Graham, 1995; Gerig, Styner, Jones, Weinberger, & Lieberman, 2001; Styner, Gerig, Lieberman, Jones, & Weinberger, 2003)) that aims to detect or localize shape changes between groups of 3D objects. This chapter describes a general shape-based 3D data mining framework for morphometric pattern discovery.

BACKGROUND

The challenges of morphometric pattern discovery are twofold: (1) How to describe a 3D shape and extract shape features; and (2) how to use shape features for pattern analysis to find discriminative regions. Several shape descriptors have been proposed for extracting shape features, including landmark-based descriptors (Bookstein, 1997; Cootes, Taylor, Cooper, & Graham, 1995), deformation fields (Csernansky et al., 1998), distance transforms (Golland, Grimson, Shenton, & Kikinis, 2001), medial axes (Styner, Gerig, Lieberman, Jones, & Weinberger, 2003), and parametric surfaces (Gerig, Styner, Jones, Weinberger, & Lieberman, 2001). Using these features, researchers have developed different pattern analysis techniques for discovering morphometric patterns, including linear discriminant analysis (Csernansky et al., 1998), support vector machines (Golland, Grimson, Shenton, & Kikinis, 2001), principal component analysis (Saykin et al., 2003), and random field theory (Chung et al., 2005).

This chapter describes a general surface-based computational framework for mining 3D objects to localize shape changes between groups. The spherical harmonic (SPHARM) method is employed for surface modeling, where several important shape analysis issues are addressed, including spherical parameterization, surface registration, and multi-object alignment. Two types of techniques are employed for statistical shape analysis: (1) linear classifiers based on a point distribution model, and (2) random field theory combined with heat kernel smoothing.

MAIN FOCUS

Given a set of labeled 3D objects from two distinct shape classes, our task is to identify morphometric patterns that can distinguish these two classes. An important real-life application is to detect anatomical changes due to pathology in biomedical imaging. A surface-based computational framework is presented to solve this problem in three steps: data collection and preprocessing, surface modeling for feature extraction, and pattern analysis and visualization.

Data Collection and Preprocessing

3D models can be collected using different methods including 3D digitizing, non-invasive scanning and interactive authoring. This chapter focuses on the analysis of 3D models whose surface topology is spherical. For example, in medical domain, many human organs and structures belong to this category. After performing segmentation on 3D medical scans (e.g., CT, MRI), the boundary of a structure of interest can be extracted. Since such a 3D boundary model may contain unwanted holes, a preprocessing step sometimes is required to close these 3D holes (Aktouf, Bertrand, & Perroton, 2002). An alternative approach is to perform automatic segmentation with appropriate constraints and create topologically correct results directly from images. After removing unwanted 3D holes, the surface of the 3D model has a spherical topology, which meets the requirement of our surface modeling approach.

Surface Modeling

Spherical harmonics were first used as a type of parametric surface representation for radial surfaces $r(\theta,\varphi)$ in (Ballard & Brown, 1982), where the harmonics were used as basis functions to expand $r(\theta,\varphi)$. Recently, an extended method, called SPHARM, was proposed in (Brechbuhler, Gerig, & Kubler, 1995) to model arbitrarily shaped but simply connected 3D objects, where three functions of θ and φ were used to represent a surface. SPHARM is suitable for surface comparison and can deal with protrusions and intrusions. Due to its numerous advantages such as inherent interpolation, implicit correspondence, and accurate scaling, SPHARM is employed here, requiring three processing steps: (1) spherical parameterization, (2) SPHARM expansion, and (3) SPHARM normalization.

(1) *Spherical parameterization* creates a continuous and uniform mapping from the object surface on to the unit sphere, and its result is a bijective mapping between each point v on the object surface and spherical coordinates θ and φ:

$$v(\theta,\phi) = (x(\theta,\phi), y(\theta,\phi), z(\theta,\phi))^T$$

The classical approach exploits the uniform quadrilateral structure of a voxel surface and solves a constrained optimization problem to minimize area and angle distortions of the parameterization. The approach can be applied only to voxel surfaces and not to general triangle meshes. A new algorithm CALD (Shen & Makedon, 2006) has been proposed to control both area and length distortions and make SPHARM applicable to general triangle meshes.

(2) The *SPHARM expansion* requires a spherical parameterization performed in advance. The parameterization has the form of:

$$v(\theta,\phi) = (x(\theta,\phi), y(\theta,\phi), z(\theta,\phi))^T,$$

Where $x(\theta,\varphi)$, $y(\theta,\varphi)$ and $z(\theta,\varphi)$ are three spherical functions. Spherical harmonics are a natural choice of basis functions for representing any twice-differentiable spherical function. To describe the object surface, we can expand these three spherical functions using spherical harmonics $Y_l^m(\theta,\phi)$, where $Y_l^m(\theta,\phi)$ denotes the spherical harmonic of degree l and order m. The expansion takes the form:

$$v(\theta,\phi) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} c_l^m Y_l^m(\theta,\phi),$$

Where

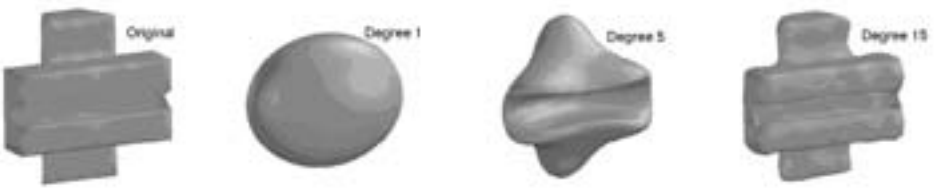
$$c_l^m = (c_{lx}^m, c_{ly}^m, c_{lz}^m)^T.$$

The coefficients c_l^m up to a user-desired degree can be estimated by solving a linear system. The object surface can be reconstructed using these coefficients, and using more coefficients leads to a more detailed reconstruction (Figure 1).

(3) *SPHARM normalization* creates a shape descriptor (*i.e.*, excluding translation, rotation, and scaling) from a normalized set of SPHARM coefficients, which are comparable across objects. A typical approach is as follows: (1) Rotation invariance is achieved by aligning the degree one ellipsoid; (2) scaling invariance is achieved by dividing all the coefficients by a scaling factor; (3) ignoring the degree zero coefficient results in translation invariance.

Using the degree one ellipsoid for establishing surface correspondence and aligning objects may not be sufficient in many cases (e.g., the ellipsoid becomes a sphere). A more general method for establishing surface correspondence is to minimize the mean squared distance between two corresponding surfaces (Huang et al., 2005).

Figure 1. The first picture shows an object surface. The second, third and fourth pictures show its SPHARM reconstructions using coefficients up to degrees 1, 5 and 15, respectively.



The SPHARM technique described above examines shape configurations containing only a single object. In some cases, it is useful to consider multi-object shape configurations so that spatial relation among objects can be extracted as additional features for structural analysis. To create a shape descriptor for a multi-object configuration, one can use the degree one ellipsoid for establishing surface correspondence and then use a quaternion-based algorithm to align surfaces in the object space (Shen, Makedon, & Saykin, 2004).

In both single-object and multi-object cases, the SPHARM coefficients are converted to a dual landmark representation for the convenience of processing. Landmarks are the points of the sampled surface: equal processing of each part of the surface can be ensured by choosing a homogeneous distribution of sampling points; and the sampling resolution can be decided by the size of the object. The landmark representation is an intuitive descriptor that can capture local shape characteristics and has the potential to support the identification of local discriminative patterns implied by a linear classifier. Thus, as a result of the SPHARM surface modeling, in both single-object and multi-object cases, each of the shape configurations is described by a set of normalized landmarks, which are comparable across individuals. These landmark descriptors serve as inputs to the next stage of pattern analysis and visualization.

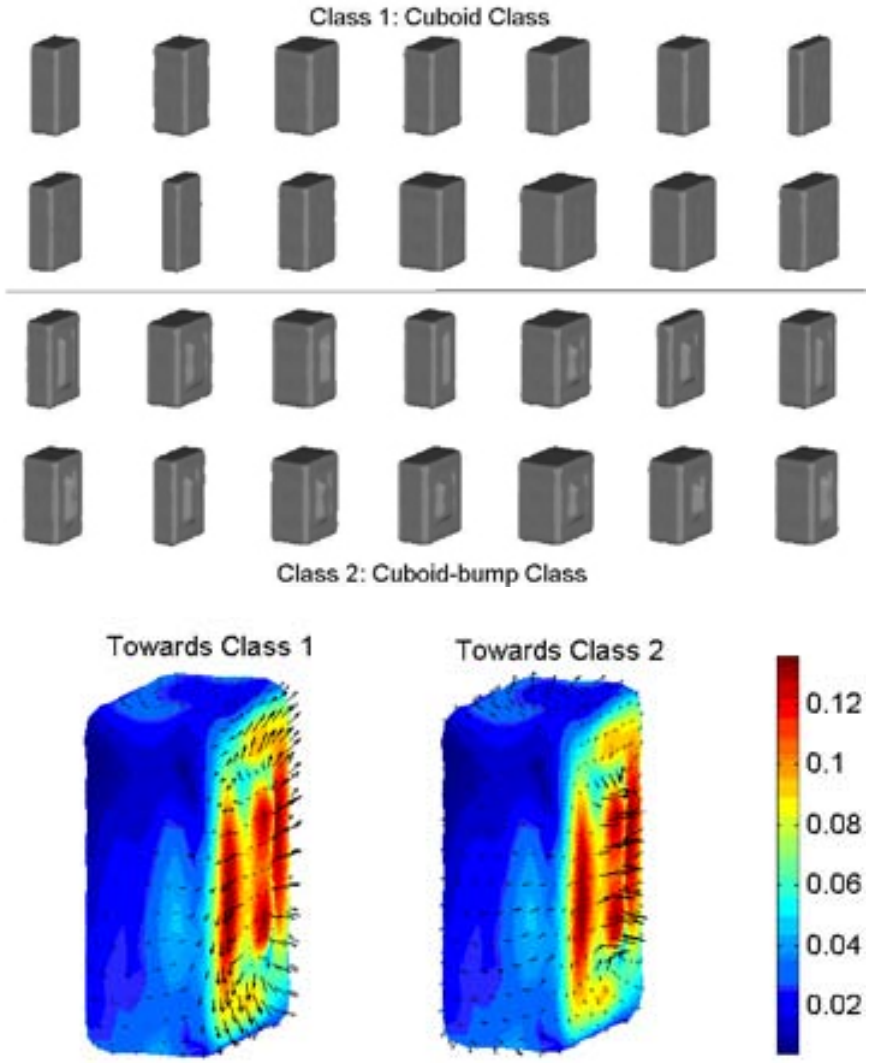
Pattern Analysis and Visualization

To use the derived landmark representation for statistical shape analysis and identification of discriminative patterns, two types of techniques are employed: (1) linear classifiers based on a point distribution model, and (2) random field theory combined with heat kernel smoothing.

Without loss of generality, a single-object case is used to describe the linear classifier approach. Assume that the icosahedron subdivision level 3 is used for surface sampling. Thus, each surface contains n = 642 landmarks and 3n = 1926 feature elements (each landmark is described by its x, y, z coordinates). Principal component analysis (PCA) is applied for dimensionality reduction. This involves eigenanalysis of the covariance matrix Σ of the data: $\Sigma P = PD$; where the columns of P hold eigenvectors, and the diagonal matrix D holds eigenvalues of Σ. The eigenvectors in P can be ordered decreasingly according to respective eigenvalues, which are proportional to the variance explained by each eigenvector. Now any shape $\mathbf{x}$ in the data can be obtained using $\mathbf{x} = \mathbf{x} + P\mathbf{b}$; where $\mathbf{b}$ is a vector containing the components of $\mathbf{x}$ in basis P, which are called *principal components* (PCs). Since eigenvectors are orthogonal, $\mathbf{b}$ can be obtained using $\mathbf{b} = P^T(\mathbf{x} - \mathbf{x})$. Given a dataset of m objects, the first m-1 PCs are enough to capture all the data variance. Thus, $\mathbf{b}$ becomes an m-1 element vector, which can be thought of a more compact representation of the shape $\mathbf{x}$. This model is a point distribution model (PDM) (Cootes, Taylor, Cooper, & Graham, 1995). PDM is applied to the data set to obtain a $\mathbf{b}$ (referred to as a feature vector hereafter) for each shape.

Feature selection may improve classification accuracy by reducing the number of parameters that need to be estimated. In this study, features are PCs, and some PCs are more useful than others for classification, but not necessarily matching the ordering of the variance amounts they explain. Our feature selection scheme selects the first n features according to a certain ordering of PCs, where varying values of n are considered. A typical ordering scheme can be that PCs are ordered by p-value associated with t-test applied to the training set, increasingly.

Figure 2. The left plot shows two classes of rectangular surfaces: the top two rows show 14 surfaces in Class 1, while the bottom two rows show 14 surfaces in Class 2. Note that surfaces in Class 2 have a centered bump on one face. The right plot shows the discriminative patterns mapped onto the mean surface. Red colors indicate the surface location has more discriminative power while blue colors indicate less. The vectors show the deformation directions towards Class 1 (top) and Class 2 (bottom).



Linear classifiers are employed on selected features for classification. Linear techniques are simple and well-understood. Once they succeed in real applications, the results can then be interpreted more easily than those derived from complicated techniques. Typical linear classifiers include Fisher's linear discriminant (FLD) and linear support vector machines (SVMs). FLD projects a training set (consisting of *c* classes) onto *c-1* dimensions such that the ratio of between-class and within-class variability is maximized, which occurs when the FLD projection places different classes into distinct and tight clumps. A linear classifier corresponds to a decision hyperplane separating different classes. The margin is defined as the distance from the decision hyperplane to the closest training set exemplar. The aim in training a linear SVM is to find the separating hyperplane with the largest margin; the expectation is that the larger the margin, the better the generalization of the classifier. The FLD and SVM classifiers have been integrated with SPHARM to form a new frame-

work for 3D surface object classification (Shen, Ford, Makedon, & Saykin, 2004). Applying it to hippocampal data in schizophrenia achieved best cross-validation accuracies of 93% (n=56), competitive with the best prior results.

The discriminative pattern captured by this classifier can be visualized as follows. Let us apply PCA and FLD (possibly with feature selection) to a landmark-based shape set, we get a discriminative value v for each shape $\mathbf{x}$: $v = \mathbf{x}_\delta^T * B_{pca} * B_{fld} = \mathbf{x} * \mathbf{w}$; where $\mathbf{x}_\delta = \mathbf{x} - \mathbf{x}$ is the deformation of $\mathbf{x}$ from the mean shape $\mathbf{x}$, B_{pca} consists of a subset of eigenvectors, depending on which PCs are selected, and B_{fld} is the corresponding FLD basis. Thus $\mathbf{w}$ is a column vector that weights the contribution of each deformation element in $\mathbf{x}_\delta$ to v. We can map these weights onto the mean surface to show significant discriminative regions (Figure 2).

Besides using linear classifiers for pattern analysis and visualization, an alternative approach performs statistical inference directly on the surface signals. For each landmark, the local shape change of an individual is defined as the distance from the mean along its normal direction. In order to increase the signal-to-noise ratio (SNR), Gaussian kernel smoothing is desirable in many statistical analyses. Since the geometry of an object surface is non-Euclidean, we employ heat kernel smoothing, which generalizes Gaussian kernel smoothing to arbitrary Riemannian manifolds (Chung et al., 2005). The heat kernel smoothing is implemented by constructing the kernel of a heat equation on manifolds that is isotropic in the local conformal coordinates. By smoothing the data on the hippocampal surface, the SNR will increase and it will be easier to localize the shape changes.

To perform statistical inference directly on the surface, surface signals are modeled as Gaussian random fields. This theoretical model assumption can be checked using either Lilliefors test or quantile-quantile plots for the shape data. Detecting the region of statistically significant shape changes can be done via thresholding the maximum of the *t* random field defined on the surface. The *p* value of the local maxima of the *t* field will give a conservative threshold. See (Chung et al., 2005) for more details on how to create a corrected *p* value map using a *t* value map and other related information. This technique has been successfully applied to identifying hippocampal shape changes in mild cognitive impairment (Shen et al., 2005).

FUTURE TRENDS

In the near future, we expect that shape-based retrieval, analysis, mining of 3D models will become a very important research area in both data mining and related application domains including computer vision, graphics, biomedical imaging and bioinformatics. There are many interesting and open problems that need to be further explored. For example, automatic segmentation of objects from 3D images remains as a very difficult task and a time consuming process; and further technological advances can help collect useful 3D models more efficiently. The scalability of the existing SPHARM method is limited. To facilitate the mining of large 3D models, there is a need to develop more scalable spherical parameterization and SPHARM expansion algorithms to model large surfaces (e.g., brain cortex) accurately and efficiently. Previous studies on morphometric pattern discovery involve relatively small data sets. With the fast growth of 3D model collections, more efficient and scalable shape-based pattern analysis techniques need to be developed to facilitate mining large-scale collections of 3D models. Morphometric pattern discovery mostly has impact in biomedical imaging and biotechnology. However the shape analysis techniques developed in this domain can be extended to solve other problems such as shape-based retrieval. Shape-based retrieval is in fact a very important research topic which has a wide range of applications in different areas such as bioinformatics (find similar proteins), graphics (find similar models from the web repository), and industrial engineering (find similar CAD/CAM models from engineering databases).

CONCLUSION

We has described a computational framework for data mining of 3D objects using shape features, and the goal is to localize shape changes between groups of 3D objects. The spherical harmonic (SPHARM) method is employed for surface modeling. We discuss a classical spherical parameterization method designed for voxel surfaces as well as a new approach that works for general triangle meshes. Besides using the first order ellipsoid for establishing surface correspondence and aligning objects, we discuss a new and more general method for establishing surface correspondence that

aims to minimize the mean squared distance between two corresponding surfaces. Surface registration issues are addressed for both single-object cases and multi-object cases. Two types of techniques are employed for mining morphometric patterns: (1) linear classifiers based on a point distribution model, and (2) random field theory combined with heat kernel smoothing. Future trends on the topic are also discussed.

REFERENCES

Aktouf, Z., Bertrand, G., & Perroton, L. (2002). A three-dimensional holes closing algorithm. *Pattern Recognition Letters, 23*, 523–531.

Ballard, D. H., & Brown, C. M. (1982). *Computer Vision*: Prentice Hall.

Bookstein, F. L. (1997). Shape and the information in medical images: A decade of the morphometric synthesis. *Computer Vision and Image Understanding, 66*(2), 97-118.

Brechbuhler, C., Gerig, G., & Kubler, O. (1995). Parametrization of closed surfaces for 3D shape description. *Computer Vision and Image Understanding, 62*(2), 154-170.

Chung, M. K., Robbins, S., Dalton, K. M., Davidson, R. J., Alexander, A. L., & Evans, A. C. (2005). Cortical thickness analysis in autism via heat kernel smoothing *NeuroImage, 25*, 1256-1265.

Cootes, T. F., Taylor, C. J., Cooper, D. H., & Graham, J. (1995). Active shape models-their training and application. *Computer Vision and Image Understanding, 61*, 38-59.

Csernansky, J. G., Joshi, S., Wang, L., Halleri, J. W., Gado, M., Miller, J. P., et al. (1998). Hippocampal morphometry in schizophrenia by high dimensional brain mapping. *Proc. National Academy of Sciences USA 95*, 11406-11411.

Funkhouser, T., Min, P., Kazhdan, M., Chen, J., Halderman, A., Dobkin, D., et al. (2003). A search engine for 3D models. *ACM Transactions on Graphics, 22*(1), 83-105.

Gerig, G., Styner, M., Jones, D., Weinberger, D., & Lieberman, J. (2001). *Shape analysis of brain ventricles using SPHARM.* Paper presented at the IEEE Workshop on Mathematical Methods in Biomedical Image Analysis.

Golland, P., Grimson, W. E. L., Shenton, M. E., & Kikinis, R. (2001). *Deformation analysis for shape based classification.* Paper presented at the 17th International Conference on Information Processing and Medical Imaging (IPMI 2001).

Huang, H., Shen, L., Zhang, R., Makedon, F., Hettleman, B., & Pearlman, J. (2005). *Surface Alignment of 3D Spherical Harmonic Models: Application to Cardiac MRI Analysis.* . Paper presented at the MICCAI 2005, 8th International Conference on Medical Image Computing and Computer Assisted Intervention, Palm Springs, California, USA.

Saykin, A. J., Flashman, L. A., McHugh, T., Pietras, C., McAllister, T. W., Mamourian, A. C., et al. (2003, March 29 - April 2, 2003). *Principal Components Analysis of Hippocampal Shape in Schizophrenia.* Paper presented at the International Congress on Schizophrenia Research, Colorado Springs, Colorado, USA.

Shen, L., Ford, J., Makedon, F., & Saykin, A. J. (2004). A surface-based approach for classification of 3D neuroanatomic structures. *Intelligent Data Analysis, An International Journal, 8*(5), 519-542.

Shen, L., & Makedon, F. (2006). Spherical Mapping for Processing of 3-D Closed Surfaces. *Image and Vision Computing, 24*(7), 743-761.

Shen, L., Makedon, F., & Saykin, A. J. (2004, Feb. 14–19). *Shape-based discriminative analysis of combined bilateral hippocampi using multiple object alignment.* Paper presented at the Medical Imaging 2004: Image Processing, San Diego, CA.

Shen, L., Saykin, A., McHugh, T., West, J., Rabin, L., Wishart, H., et al. (2005, July 21-26). *Morphometric analysis of 3D surfaces: Application to hippocampal shape in mild cognitive impairment.* Paper presented at the CVPRIP 2005: 6th Int. Conf. on Computer Vision, Pattern Recognition and Image Processing, Salt Lake City, Utah.

Styner, M., Gerig, G., Lieberman, J., Jones, D., & Weinberger, D. (2003). Statistical shape analysis of neuroanatomical structures based on medial models *Medical Image Analysis, 7*(3), 207-220.

KEY TERMS

3D Data Mining: The process of automatically searching large volumes of 3D data for hidden morphometric patterns.

Discriminative Patterns: Patterns that can distinguish two or more sets of objects.

Morphometric Patterns: Patterns that are related to or contain shape information, such as a description of the mean and variability of a shape population.

Morphometrics: The branch of mathematics studying the metrical and statistical properties of shapes and shape changes of geometric objects.

Parametric Surfaces: Surfaces that are defined by functions based on an underlying parameterization.

Shape: All the geometrical information that is invariant to location, scale and rotation.

Shape Alignment: The adjustment of a set of geometric configurations in relation to one another so that their shapes can be compared under a common coordinate system.

Shape Description: A set of variables that describe the shape of a geometric configuration.

Spherical Harmonics: An orthogonal set of solutions to Laplace's equation represented in a system of spherical coordinates.

Surface Parameterization: The process of creating a one-to-one mapping from some parameter domain to a surface.

Mining Chat Discussions

Stanley Loh
Catholic University of Pelotas, Brazil
Lutheran University of Brazil, Brazil

Daniel Licthnow
Catholic University of Pelotas, Brazil

Thyago Borges
Catholic University of Pelotas, Brazil

Tiago Primo
Catholic University of Pelotas, Brazil

Rodrigo Branco Kickhöfel
Catholic University of Pelotas, Brazil

Gabriel Simões
Catholic University of Pelotas, Brazil

Gustavo Piltcher
Catholic University of Pelotas, Brazil

Ramiro Saldaña
Catholic University of Pelotas, Brazil

INTRODUCTION

According to Nonaka & Takeuchi (1995), the majority of the organizational knowledge comes from interactions between people. People tend to reuse solutions from other persons in order to gain productivity.

When people communicate to exchange information or acquire knowledge, the process is named *Collaboration*. Collaboration is one of the most important tasks for innovation and competitive advantage within *learning organizations* (Senge, 2001). It is important to record knowledge to later reuse and analysis. If knowledge is not adequately recorded, organized and retrieved, the consequence is re-work, low productivity and lost of opportunities.

Collaboration may be realized through synchronous interactions (e.g., exchange of messages in a chat), asynchronous interactions (e.g., electronic mailing lists or forums), direct contact (e.g., two persons talking) or indirect contact (when someone stores knowledge and others can retrieve this knowledge in a remote place or time).

In special, chat rooms are becoming important tools for collaboration among people and knowledge exchange. Intelligent software systems may be integrated into chat rooms in order to help people in this collaboration task. For example, systems can identify the theme being discussed and then offer new information or can remember people of existing information sources. This kind of systems is named recommender systems.

Furthermore, chat sessions have implicit knowledge about what the participants know and how they are viewing the world. Analyzing chat discussions allows understanding what people are looking for and how people collaborates one with each other. Intelligent software systems can analyze discussions in chats to extract knowledge about the group or about the subject being discussed.

Mining tools can analyze chat discussions to understand what is being discussed and help people. For example, a recommender system can analyze textual messages posted in a web chat, identify the subject of the discussion and then look for items stored in a Digital Library to recommend individually to each participant of the discussion. Items can be electronic documents, web pages and bibliographic references stored in a digital library, past discussions and authorities (people with expertise in the subject being discussed). Besides that, mining tools can analyze the whole discussion to map the knowledge exchanged among the chat participants.

The benefits of such technology include supporting learning environments, knowledge management efforts within organizations, advertisement and support to decisions.

BACKGROUND

Some works has investigated the analysis of online discussions. Brutlag and Meek (2000) have studied the identification of themes in e-mails. The work compares the identification by analyzing only the subject of the e-mails against analyzing the message bodies. One conclusion is that e-mail headers perform so well

as message bodies, with the additional advantage of reducing the number of features to be analyzed.

Busemann et al. (2000) investigated the special case of messages registered in call centers. The work proved possible to identify themes in this kind of message, although the informality of the language used in the messages. This informality causes mistakes due to jargons, misspellings and grammatical inaccuracy.

The work of Durbin et al. (2003) has shown possible to identify affective opinions about products and services in e-mails sent by customers, in order to alert responsible people or to evaluate the organization and customers' satisfaction. Furthermore, the work identifies the intensity of the rating, allowing the separation of moderate or intensive opinions.

Tong (2001) investigated the analysis of online discussions about movies. Messages represent comments about movies. This work proved to be feasible to find positive and negative opinions, by analyzing key or cue words. Furthermore, the work also extracts information about the movies, like directors and actors, and then examines opinions about these particular characteristics.

The only work found in the scientific literature that analyzes chat messages is the one from Kahn et al. (2002). They apply mining techniques over chat messages in order to find social interactions among people. The goal is to find who is related to whom inside a specific area, by analyzing the exchange of messages in a chat and the subject of the discussion.

MAIN THRUST

Following, the chapter explains how messages can be mined, how recommendations can be made and how the whole discussion (an entire chat session) can be analyzed.

Identifying Themes in Chat Messages

To provide people with useful information during a collaboration session, the system has to identify what is being discussed. Textual messages sent by the users in the chat can be analyzed for this purpose. Texts can lead to the identification of the subject discussed because the words and the grammar present in the texts represent knowledge from people, expressed in written formats (Sowa, 2000).

An ontology or thesaurus can be used to help to identify cue words for each subject. The ontology or thesaurus has concepts of a domain or knowledge area, including relations between concepts and the terms used in written languages to express these concepts (Gilchrist, 2003). The ontology can be created by machine learning methods (supervised learning), where human experts select training cases for each subject (for example, texts of positive and negative examples) and an intelligent software system identifies the keywords that define each subject. The TFIDF method from Salton & McGill (1983) is the most used in this kind of task.

If considering that the terms that compose the messages compose a bag of words (have no difference in importance), probabilistic techniques can be used to identify the subject. By other side, natural language processing techniques can identify syntactic elements and relations, then supporting more precise subject identification.

The identification of themes should consider the context of the messages to determine if the concept identified is really present in the discussion. A group of messages is better to infer the subject than a single message. That avoids misunderstandings due to words ambiguity and use of synonyms.

Making Recommendations in a Chat Discussion

A recommender system is a software whose main goal is to aid in the social collaborative process of indicating or receiving indications (Resnick & Varian, 1997). Recommender systems are broadly used in electronic commerce for suggesting products or providing information about products and services, helping people to decide in the shopping process (Lawrence et al., 2001) (Schafer et al., 2001). The offered gain is that people do not need to request recommendation or to perform a query over an information base, but the system decides what and when to suggest. The recommendation is usually based on user profiles and reuse of solutions.

When a subject is identified in a message, the recommender searches for items classified in this subject. Items can come from different databases. For example, a Digital Library may provide electronic documents, links to Web pages and bibliographic references.

A profile database may contain information about people, including the interest areas of each person, as well an associated degree, informing the user's

knowledge level on the subject or how much is his/her competence in the area (his/her expertise). This can be used to indicate who is the most active user in the area or who is the authority in the subject.

A database of past discussions records everything that occurs in the chat, during every discussion session. Discussions may be stored by sessions, identified by data and themes discusses and can include who participated in the session, all the messages exchanged (with a label indicating who sent it), the concept identified in each message, the recommendations made during the session for each user and documents downloaded or read during the session. Past discussions may be recommended during a chat session, remembering the participants that other similar discussions have already happened. This database also allows users to review the whole discussion later after the session. The great benefit is that users do not re-discuss the same question.

Mining a Chat Session

Analyzing the themes discussed in a chat session can bring an important overview of the discussion and also of the subject. Statistical tools applied over the messages sent and the subjects identified in each message can help users to understand which were the themes more discussed. Counting the messages associated with each subject, it is possible to infer the central point of the discussion and the peripheral themes.

The list of subjects identified during the chat session compose an interesting order, allowing users to analyze the path followed the participants during the discussion. For example, it is possible to observe which was the central point of the discussion, whether the discussion deviated from the main subject and whether the subjects present in the beginning of the discussion were also present at the end. The coverage of the discussion may be identified by the number of different themes discussed.

Furthermore, this analysis allows identifying the depth of the discussion, that is, whether more specific themes were discussed or whether the discussion occurred superficially at a higher conceptual level. The analysis can also map the coverage of the discussion: how many different subjects were discussed.

Analyzing the messages sent by every participant allows determining the degree of participation of each person in the discussion: who participated more and who did less. Furthermore, it is possible to observe which are the interesting areas for each person and in someway to determine the expertise of the group and of the participants (which are the areas where the group is more competent).

Association techniques can be used to identify correlations between themes or between themes and persons. For example, it is possible to find that some theme is present always when other theme is also present or to find that every discussion where some person participated had a certain theme as the principal.

FUTURE TRENDS

Recommender systems are still an emerging area. There are some doubts and open issues. For example, whether is good or bad to recommend items already suggested in past discussions (re-recommend, as if remembering the person). Besides that it is important to analyze the level of the participants in order to recommend only basic or advanced items.

Collaborative filtering techniques can be used to recommend items already seen by other users (Resnick et al., 1994; Terveen and Hill, 2001). Grouping people with similar characteristics allows to cross recommended items, for example, offer documents read by one person to others.

In the same way, software systems can capture relevance feedback from users to narrow the list of recommendations. Users should read some items of the list and rate them, so that the system can use this information to eliminate items from the list or to reorder the items in a new ranking.

The context of the messages needs to be more studied. To infer the subject being discussed, the system can analyze a group of messages, but it is necessary to determine how many (a fixed number or all messages sent in the past N minutes ?).

An orthographic corrector is necessary to clean the messages posted to the chat. Lots of linguistic mistakes are expected since people are using chats in a hurry, with little attention to the language, without revisions and in an informal way. Furthermore, the text mining tools must analyze special signs like novel abbreviations, emoticons and slang expressions. Special words may be added to the domain ontology in order to hold the differences in the language.

CONCLUSION

An example of such a system discussed in this chapter is available in http://gpsi.ucpel.tche.br/sisrec. Currently, the system uses a domain ontology for Computer Science, but others can be used. Similarly, the current digital library only has items related to Computer Science.

The recommendation system facilitates the organizational learning because people receive suggestions of information sources during online discussions. The main advantage of the system is to free the user of the burden to search information sources during the online discussion. Users do not have to choose attributes or requirements from a menu of options, in order to retrieve items of a database; the system decides when and what information to recommend to the user. This proactive approach is useful for non-experienced users that receive hits about what to read in a specific subject. User's information needs are discovered naturally during the conversation.

Furthermore, when the system indicates people who are authorities in each subject, naïve users can meet these authorities for getting more knowledge.

Other advantage of the system is that part of the knowledge shared in the discussion can be made explicit through the record of the discussion for future retrieval. Besides that, the system allows the posterior analysis of each discussion, presenting the subjects discussed, the messages exchanged, the items recommended and the order in which the subjects were discussed.

An important feature is the statistical analysis of the discussion, allowing understanding the central point, the peripheral themes, the order of the discussion, its coverage and depth.

The benefit of mining chat sessions is of special interest for Knowledge Management efforts. Organizations can store tacit knowledge formatted as discussions. The discussions can be retrieved, so that knowledge can be reused. In the same way, the contents of a Digital Library (or Organizational Memory) can be better used through recommendations. People do not have to search for contents neither to remember items in order to suggest to others. Recommendations play this role in a proactive way, examining what people are discussing and users' profiles and selecting interesting new contents.

In special, such systems (that mine chat sessions) can be used in e-learning environments, supporting the construction of knowledge by individuals or groups. Recommendations help the learning process, suggesting complementary contents (documents and sites stored in the Digital Library). Recommendations also include authorities in topics being discussed, that is, people with high degree of knowledge.

ACKNOWLEDGMENTS

This research group is partially supported by CNPq, an entity of the Brazilian government for scientific and technological development.

REFERENCES

Brutlag, J. D. & Meek, C. (2000). Challenges of the email domain for text classification. In: *Proceedings of the 7th International Conference on Machine Learning* (ICML 2000), Stanford University, Standord, CA, USA, 103-110.

Busemann, S., Schmeier, S., Arens, R.G. (2000) Message classification in the call center. In: *Proceedings of the Applied Natural Language Processing Conference* – ANLP'2000, Seattle, WA, 159-165.

Durbin, S. D., Richter, J. N., Warner, D. (2003). A system for affective rating of texts. In *Proceedings of the 3rd Workshop on Operational Text Classification, 9th ACM International Conference on Knowledge Discovery and Data Mining* (KDD-2003), Washington, DC.

Khan, F. M., Fisher, T. A., Shuler, L., Wu, T., Pottenger, W. M. (2002). Mining chat-room conversations for social and semantic interactions. Technical Report LU-CSE-02-011, Lehigh University, Bethlehem, Pennsylvania, USA.

Gilchrist, A. (2003). Thesauri, taxonomies and ontologies – an etymological note. *Journal of Documentation*, 59 (1), 7-18.

Lawrence, R. D. et al. (2001). Personalization of supermarket product recommendations. *Journal of Data Mining and Knowledge Discovery*, 5 (1/2), 11-32.

Nonaka, I. & Takeuchi, T. (1995). *The knowledge-creating company*: how japanese companies create the dynamics of innovation. Cambridge: Oxford University Press.

Resnick, P. et al. (1994). GroupLens: an open architecture for collaborative filtering of Netnews. In: *Proceedings of the Conference on Computer Supported Coooperative Work*, 175-186.

Resnick, P. & Varian, H. (1997). Recommender systems. *Communications of the ACM*, 40 (3), 56-58.

Salton, G. & McGill, M. J. (1983). *Introduction to modern information retrieval*. New York: McGraw-Hill.

Schafer, J. B. et al. (2001). E-commerce recommendation applications. *Journal of Data Mining and Knowledge Discovery*, 5 (1/2), 115-153.

Senge, P. M. (2001). *The fifth discipline:* the art and practice of the learning organization. 9th ed. São Paulo: Best Seller (in portuguese)

Sowa, John F. (2000). *Knowledge representation*: logical, philosophical, and computational foundations. Pacific Grove, CA: Brooks/Cole Publishing Co.

Terveen, L. & Hill, W. (2001). Human-computer collaboration in recommended systems. In: J. CARROLL (editor). *Human computer interaction in the new millennium*. Boston: Addison-Wesley.

Tong, R. (2001). Detecting and tracking opinions in online discussions. In: Proceedings of the Workshop on Operational Text Classification, SIGIR, New Orleans, Louisiana, USA

KEY TERMS

Chat: A software system that enables real-time communication among users through the exchange of textual messages.

Collaboration: The process of communication among people with the goal of sharing information and knowledge.

Digital Library: A set of electronic resources (usually documents) combined with a software system which allows storing, organizing and retrieving the resources.

Knowledge Management: Systems and methods for storing, organizing and retrieving explicit knowledge.

Mining: The application of statistical techniques to infer implicit patterns or rules in a collection of data, in order to discover new and useful knowledge.

Ontology: A formal and explicit definition of concepts (classes or categories) and their attributes and relations.

Recommender System: A software system that makes recommendations to a user, usually analyzing the user's interest or need.

Recommendations: Results of the process of providing useful resources to a user, like products, services or information.

Text Mining: The process of discovering new information analyzing textual collections.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 758-762, copyright 2005 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

Mining Data Streams

Tamraparni Dasu
AT&T Labs, USA

Gary Weiss
Fordham University, USA

INTRODUCTION

When a space shuttle takes off, tiny sensors measure thousands of data points every fraction of a second, pertaining to a variety of attributes like temperature, acceleration, pressure and velocity. A data gathering server at a networking company receives terabytes of data a day from various network elements like routers, reporting on traffic throughput, CPU usage, machine loads and performance. Each of these is an example of a data stream. Many applications of data streams arise naturally in industry (networking, e-commerce) and scientific fields (meteorology, rocketry).

Data streams pose three unique challenges that make them interesting from a data mining perspective.

1. **Size:** The number of measurements as well as the number of attributes (variables) is very large. For instance, an IP network has thousands of elements each of which collects data every few seconds on multiple attributes like traffic, load, resource availability, topography, configuration and connections.
2. **Rate of accumulation:** The data arrives very rapidly, like "water from a fire hydrant". Data storage and analysis techniques need to keep up with the data to avoid insurmountable backlogs.
3. **Data transience:** We get to see the raw data points at most once since the volumes of the raw data are too high to store or access.

BACKGROUND

Data streams are a predominant form of information today, arising in areas and applications ranging from telecommunications, meteorology and sensor networks, to the monitoring and support of e-commerce sites. Data streams pose unique analytical, statistical and computing challenges that are just beginning to be addressed. In this chapter we give an overview of the analysis and monitoring of data streams and discuss the analytical and computing challenges posed by the unique constraints associated with data streams.

There are a wide variety of analytical problems associated with mining and monitoring data streams, such as:

1. Data reduction,
2. Characterizing constantly changing distributions and detecting changes in these distributions,
3. Identifying outliers, tracking rare events and anomalies,
4. "Correlating" multiple data streams,
5. Building predictive models,
6. Clustering and classifying data streams, and
7. Visualization.

As innovative applications in on-demand entertainment, gaming and other areas evolve, new forms of data streams emerge, each posing new and complex challenges.

MAIN FOCUS

The data mining community has been active in developing a framework for the analysis of data streams. Research is focused primarily in the field of computer science, with an emphasis on computational and database issues. Henzinger, Raghavan & Rajagopalan (1998) discuss the computing framework for maintaining aggregates from data using a limited number of passes. Domingos & Hulten (2001) formalize the challenges, desiderata and research issues for mining data streams. Collection of rudimentary statistics for data streams is addressed in Zhu & Sasha (2002) and Babcock, Datar, Matwani & O'Callaghan (2003). Clustering (Aggarwal, Han, Wang & Yu, 2003), classification, association rules (Charikar, Chen & Farach-

Colton, 2002) and other data mining algorithms have been considered and adapted for data streams.

Correlating multiple data streams is an important aspect of mining data streams. Guha, Gunopulous & Koudas (2003) have proposed the use of singular value decomposition (SVD) approaches (suitably modified to scale to the data) for computing correlations between multiple data streams.

A good overview and introduction to data stream algorithms and applications from a database perspective is found in Muthukrishnan (2003). Aggarwal (2007) has a comprehensive collection of work in the computer science field on data streams. In a similar vein, Gaber (2006) maintains a frequently updated website of research literature and researchers in data streams.

However, there is not much work in the statistical analysis of data streams. Statistical comparison of signatures of telecommunication users was used by Cortes & Pregibon (2001) to mine large streams of call detail data for fraud detection and identifying social communities in a telephone network. Papers on change detection in data streams (Ben-David, Gehrke & Kifer, 2004; Dasu, Krishnan, Venkatasubramanian & Yi, 2006) use statistical approaches of varying sophistication. An important underpinning of statistical approaches to data mining is density estimation, particularly histogram based approaches. Scott (1992) provides a comprehensive statistical approach to density estimation, with recent updates included in Scott & Sain (2004). A tutorial by Urbanek & Dasu (2007) sets down a statistical framework for the rigorous analysis of data streams with emphasis on case studies and applications. Dasu, Koutsofios & Wright (2007) discuss application of statistical analysis to an e-commerce data stream. Gao, Fan, Han & Yu (2007) address the issue of estimating posterior probabilities in data streams with skewed distributions.

Visualization of data streams is particularly challenging, from the three perspectives dimensionality, scale and time. Wong, Foote, Adams, Cowley & Thomas (2003) present methods based on multi dimensional scaling. Urbanek & Dasu (2007) present a discussion of viable visualization techniques for data streams in their tutorial.

Data Quality and Data Streams

Data streams tend to be dynamic and inherently noisy due to the fast changing conditions.

An important but little discussed concern with data streams is the quality of the data. Problems could and do arise at every stage of the process.

Data Gathering: Most data streams are generated automatically. For instance, a router sends information about packets at varying levels of detail. Similarly an intrusion detection system (IDS) automatically generates an alarm on a network when a predefined rule or condition is met. The data streams change when the rule settings are changed either intentionally by an operator or due to some software glitch. In either case, there is no documentation of the change to alert the analyst that the data stream is no longer *consistent* and *can not be interpreted* using the existing data definitions. Software and hardware components fail on occasion leading to gaps in the data streams (*missing or incomplete data*).

Data Summarization: Due to the huge size and rapid accumulation, data streams are usually summarized for storage -- for instance using 5-minute aggregates of number of packets, average CPU usage, and number of events of a certain type in the system logs. However, the average CPU usage might not reflect abnormal spikes. Or, a rare but catastrophic event might be unnoticed among all the other types of alarms. The trade-off between data granularity and aggregation is an important one. There has been much interest in representing data streams using histograms and other distributional summaries (Guha, Koudas & Shim, 2001) but largely for univariate data streams. Options for multivariate data streams and the use of sufficient statistics (Moore, 2006) for building regression type models for data streams are explored in Dasu, Koutsofios & Wright (2007).

Data Integration: Creating a comprehensive data set from multiple data sources always poses challenges. Sometimes there are no well defined join keys – only soft keys like names and addresses that can be represented in many different ways. For example, "J. Smith", "John Smith" and "John F. Smith" might be different variations of the same entity. Disambiguation is not easy. One data source might contain only a fraction of the entities contained in the other data sources, leading to gaps in the data matrix. Data streams pose additional complexities such as synchronization of multiple streams. There are two ways the temporal aspect could be a problem. First, if the clocks that timestamp the data streams are out of step and second, if the aggregation granularity does not allow the two data streams to be synchronized in any meaningful fashion.

Data quality is of particular concern in data streams. We do not have the luxury of referring back to the raw data to validate or correct mistakes. Furthermore, data quality mistakes could get compounded rapidly due to the high rate of accumulation, resulting in a significant divergence from reality and accuracy.

A Framework

A data stream is typically a sequential set of measurements in time. In most extant literature, a data stream is univariate i.e. measures just one attribute. It is represented in a reduced, more manageable form by maintaining statistical summaries for a given slice of the data stream, called a window, where every chunk of *N* points constitutes a window.

In Figure 1, the gray dots are data points that have been processed and summarized into aggregates *{A(t-1)}*, the black dots represent the data in the current time window that will be stored as aggregates *{A(t)}* and the white dots are data points yet to be processed. Another approach to maintaining summaries is to compute cumulative summaries based on the entire history of the data stream and updating these as the data comes in.

Aggregates are typically counts, sums and higher order moments like sums of squares; extreme values like minimum and maximum; and other percentiles. It is important to select these summaries carefully since there is no further opportunity to access the raw data to either update the existing summaries or compute additional aggregates. The amount of type and kind of statistics to be maintained can be customized depending on the application and may include:

1. Characterizing the distribution of a data stream by building histograms – see Guha, Koudas & Shim (2001) and Muthukrishnan (2003).
2. Detecting changes in the distribution and updating the distribution to reflect the changes – change detection has received much attention since it plays a critical role in network monitoring, security applications and maintaining robust and reliable aggregates for data stream modeling. We will discuss this in a little greater detail later on in the context of the application.
3. Comparing two or more data streams (Cormode, Datar, Indyk & Muthukrishnan (2002)) or the same data stream at different points in time (Dasu, Krishnan, Venkatasubramanian & Yi (2006)).
4. Detecting anomalies and outliers in a data stream.
5. Discovering patterns and sequences in data stream anomalies. See Aggarwal (2006).
6. Building predictive models.

Figure 1

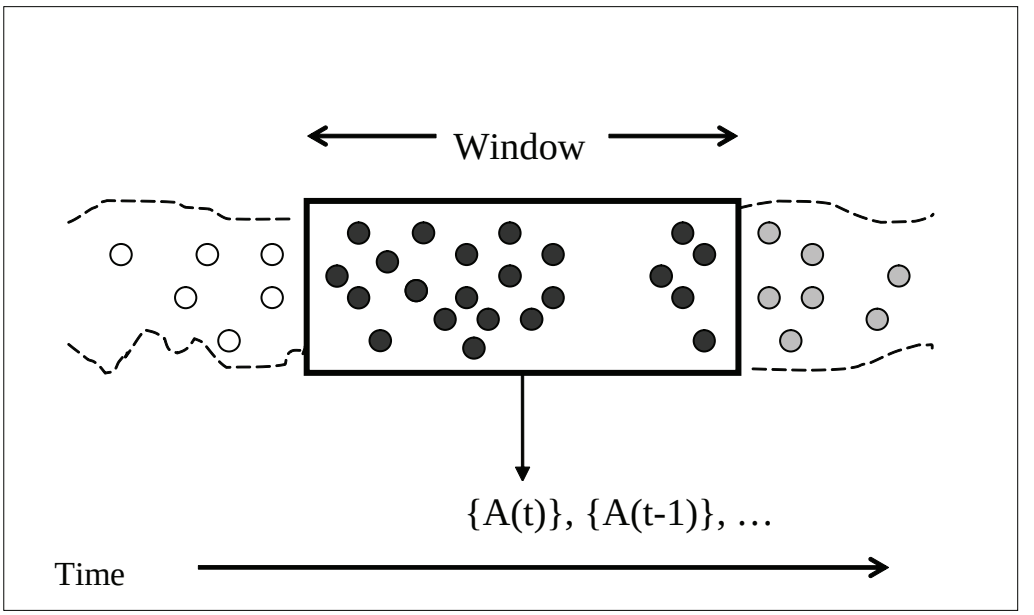
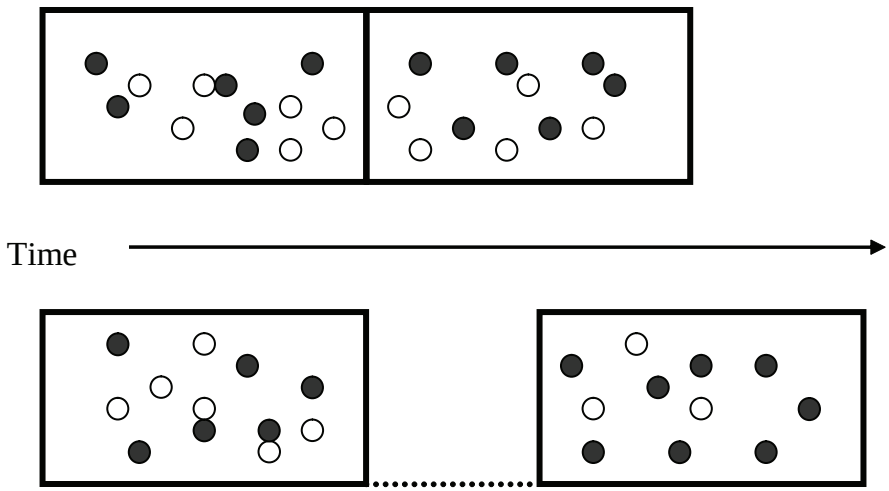


Figure 2.



An important example of the use of data stream summaries is change detection in data streams. Changes are detected by comparing data in two or more windows.

Short term changes are detected using adjacent windows that move in lock-step across time. Long term changes are detected using fixed-slide windows where one window is kept fixed while the second window moves forward in time. When a change is detected, the fixed window is moved to the most recent position and the process starts all over again. Ben-David, Gehrke & Kifer (2004) use a rank based method (Lehmann, 1974) to detect changes in the two windows. However, this can not be extended to higher dimensional data streams since there is no ordering of data in higher dimensions. Nor can it be used for categorical attributes. The method proposed by Dasu, Krishnan, Venkatasubramanian & Yi (2006) based on the Kullback-Leibler information theoretic distance measure addresses these shortcomings. We give a brief description of the basic methodology since our case study relies on this technique.

1. First, use any data partitioning scheme to "bin" the data in the two windows being compared. The partition can be predefined, based on the values of categorical attributes (e.g. gender) or intervals of continuous attributes (e.g., income), or a simple data-driven grid, based on the quantiles of individual attributes. A partition can also be induced by a model such as a clustering or classification algorithm. In our application, we use a DataSphere partition (Johnson & Dasu, 1998) that has the property that the number of bins increases linearly with the number of dimensions or attributes. It is characterized by *distance layers* that divide the data into groups of data points that are within a distance range of a given reference point such as the mean; and *directional pyramids* characterized by the direction of greatest deviation. A detailed discussion is beyond the scope of this chapter.
2. Next, we compute two histograms *H1* and *H2*, one for each of the two windows being compared. The histograms are represented by the frequency vectors

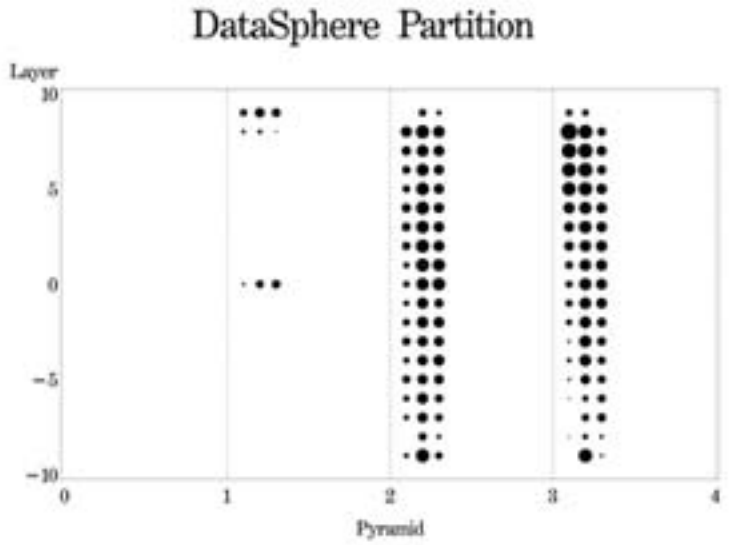
   (p1, p2, …, pB)

   and

   (q1, q2, …, qB),

   where *B* is the number of bins and *pi, qi* are the frequency counts.
3. We compare the distance between the two histograms using a range of statistical tests, like the naïve multinomial Chi-square or a similar test based on the Kullback-Leibler divergence. We use bootstrapping to simulate a sampling distribution when an exact test is not known.

Figure 3



4. Finally, we choose a desired level of confidence (e.g., 95%) and use the sampling distribution to see if the difference is significant.

The methodology is derived from the classical hypothesis testing framework of statistics. A brief discussion of statistical hypothesis testing along with the bootstrap methodology in the context of change detection is found in Dasu, Krishnan, Venkatasubramanian & Yi (2006). We use additional tests to identify regions of greatest difference. We present below an application that uses this approach.

An Application: Monitoring IP Networks

IP networks carry terabytes of data a day to enable the smooth functioning of almost every aspect of life, be it running corporations, industrial plants, newspapers, educational institutions or simpler residential tasks like exchanging e-mail or pictures. The networks are made of thousands of hardware and software components, and governed by rules called protocols that direct and regulate the data traffic. The traffic, its movement, and the functioning of the components are recorded in daunting detail in various forms. Traffic flows are recorded by netflows that specify the amount and type of data, its origin and destination, and intermediate stops if any. The topology of the network is like a dynamic road map for the data traffic and maintained in configuration tables. The functioning of the components like routers that direct the traffic is recorded in terms of resource usage. Alarms and unusual events are logged by software installed at critical points of the network. We brought together several such data sources to give us a timely and accurate picture of the state of the network.

We present below a brief analysis of a major network observed over a six week period. We are deliberately vague about the details to preserve proprietary information and some of the data distributions have been modified in a manner that does not affect the illustrative purpose of this application. For narrative convenience, we focus on three attributes of the data: the proportion of errors, the total traffic in bytes of type A, and the total traffic of type B. We use a specified week to create a baseline DataSphere partition based on the three attributes and compare the data from other weeks to the baseline. In Figure 3, we focus on a particular network device, D1. The *Y*-axis represents the distance layers, where a "negative" layer corresponds to values of deviations that are below average. The *X*-axis represents the directional pyramids. For example, a data point that has an above average proportion of errors and is

Figure 4.

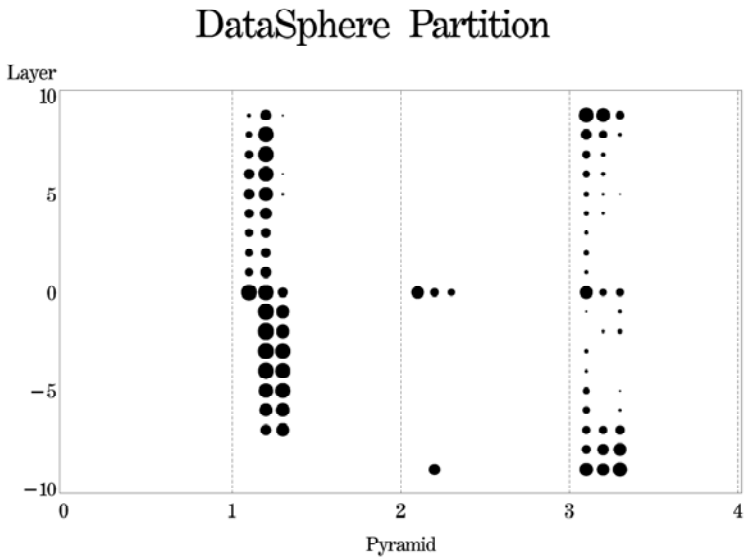


Figure 5.

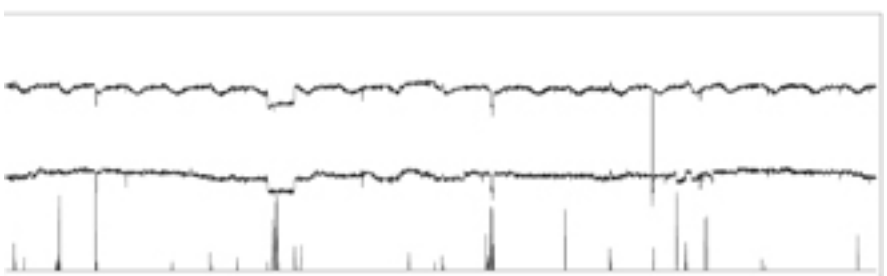
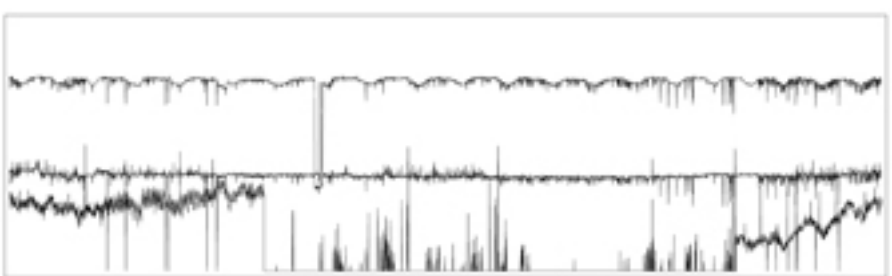


Figure 6.



most deviant with respect to that attribute will fall in pyramid 1. The more extreme the values, the greater the distance from the reference point and therefore higher the value of the distance layer. The dots are proportional to the amount of data in that bin. And the three columns within a specific pyramid range correspond to three consecutive weeks, where the middle column is the baseline week. In Figure 3, the distribution has shifted slightly in the three weeks, most notably in pyramid 3 (traffic type B) where the mass has shifted from higher values to more average values. Figure 4, describes the same information but for a different but comparable network device D2. Over the three weeks, we notice major shifts in the data mass in pyramid 1. The patterns for the error proportion are completely reversed from the first week to the third week. There is an unusually high proportion of errors which is fixed during weeks 2 and 3. This pattern is accompanied by an unusually large volume of type B traffic which returns to normal by the third week.

On examining the three individual attributes represented by the three lines in Figure 5, we see that for device D1, the individual attributes are well behaved with slight deviations from the ordinary patterns.

Figure 5 shows the same information for device D2. The erratic patterns in error proportions (bottom line in the plot) are evident, as well as the single big drop in type A traffic (top line in the plot) which corresponds to the big dot in pyramid 2, layer -8, week 2 in Figure 4.

The two dimensional "distance layer-directional pyramid" plots are a convenient and comprehensive way of displaying the distribution of the mass in the bins of the DataSphere partition, irrespective of the number of attributes. Note that line plots like the ones in Figures 5 and 6 become too numerous and overwhelming as the number of attributes increases.

In the case of the data streams above, the differences were clearly significant. In situations where the differences are more subtle, statistical tests of significance are used. See Dasu, Koutsofios & Wright (2007) for a case study that further illustrates the use of these tests.

FUTURE TRENDS

An interesting problem arises while comparing two data streams using multivariate histograms. Given the generally noisy nature of data streams, we can expect standard statistical tests to routinely declare differences. However, can we adapt the test to ignore differences in specified cells which we know a priori to be noisy and which might vary over time?

Research opportunities abound in the warehousing and querying of data streams. Aggarwal (2007) has a comprehensive collection of research articles that provide insights into the current research in the database community as well as open problems that require interdisciplinary solutions.

CONCLUSION

We have provided a brief overview of mining and monitoring data streams. Data streams are an inevitable and challenging form of information in many industrial and scientific applications, particularly the telecommunications industry. The research in this area is in its infancy and provides challenging research opportunities in managing, storing, querying and mining of data streams.

REFERENCES

Aggarwal, C. (2007). *Data Streams: Models and Algorithms*. Springer, USA.

Aggarwal, C., Han, J., Wang, J., & Yu, P. S. (2003). A Framework for Clustering Evolving Data Streams, *Proc. 2003 Int. Conf. on Very Large Data Bases*.

Babcock, B., Datar, M., Motwani, R., & L. O'Callaghan (2003). Maintaining Variance and k-Medians over Data Stream Windows. *Proceedings of the 22nd Symposium on Principles of Database Systems*.

Ben-David, S., Gehrke J., & Kifer, D. (2004). Detecting Change in Data Streams. *Proceedings of VLDB 2004.*

Bickel, P. J., & Doksum, K. A. (2001). *Mathematical statistics*, vol. 1. Prentice Hall, New Jersey.

Charikar, M., Chen K., & Farach-Colton, M. (2002). Finding Frequent Items in Data Streams. *International Colloquium on Automata, Languages, and Programming* (ICALP '02) 508--515.

Cormode, G., Datar, M, Indyk, P., & Muthukrishnan, S. (2002) Comparing data streams using Hamming

norms. In *Proceedings of the International Conference on Very Large Data Bases*, pp. 335-345.

Cortes, C., & Pregibon, D. (2001). Signature-based methods for data streams. *Data Mining and Knowledge Discovery*, *5*, 167-182.

Cover, T., & Thomas, J. (1991). *Elements of Information Theory*. John Wiley & Sons, New York.

Dasu, T., Koutsosfios, E., & Wright, J. R. (2007). A Case Study in Change Detection. In *Proc. of the International Workshop on Statistical Modelling, Barcelona, 2007.*

Dasu, T., Krishnan, S., Venkatasubramanian, S., & Yi, K. (2006). An information-theoretic approach to detecting changes in multi-dimensional data streams. *Proceedings of the 38th Symposium on the Interface of Statistics, Computing Science, and Applications (Interface '06)*, Pasadena, CA.

Domingos, P., & Hulten, G. (2001). Catching up with the data: Research issues in mining data streams. *Workshop on Research Issues in Data Mining and Knowledge Discovery, 2001*.

Efron, B., & Tibshirani, R. (1993). *An Introduction to the Bootstrap*. Chapman and Hall.

Gaber, M. M. (2006). *Mining data streams bibliography*. Retrieved from http://www.csse.monash.edu.au/~mgaber/WResources.htm

Gao, J., Fan, W., Han, J., & Yu, P. S. (2007). A general framework for mining concept-drifting data streams with skewed distributions. In *Proc. of the SIAM International Conference on Data Mining.*

Guha, S., Gunopulous D., & Koudas, N. (2003). Correlating synchronous and asynchronous data streams. In *Proc. of International Conference on Knowledge Discovery and Data Mining.*

Guha, S., Koudas, N., & Shim, K. (2001). Data-streams and histograms. In *Proc. ACM Symp. on Theory of Computing*, pp. 471-475.

Henzinger, M., Raghavan, P., & Rajagopalan, S. (1998). Computing on data streams. *Technical Note 1998-011*, Digital Systems Center, Palo Alto, CA.

Johnson, T., & Dasu, T. (1998). Comparing massive high-dimensional data sets. *Proc. 1998 KDD*, pp. 229-233..

Lehmann, E. (1974). *Nonparametric statistics: Statistical methods based on ranks*. Holden-Day.

Moore, A. (2006). *New cached-sufficient statistics algorithms for quickly answering statistical questions.* Keynote address at KDD 2006, Philadelphia.

Muthukrishnan, S. (2003). Data streams: Algorithms and Applications. *Proceedings of the fourteenth annual ACM-SIAM symposium on discrete algorithms.*

Scott, D. W. (1992). *Multivariate density estimation: Theory, practice and visualization.* John Wiley, New York.

Scott, D.W., & Sain, S.R. (2004). Multi-Dimensional Density Estimation. In C. R. Rao & E. J. Wegman (Eds.), *Handbook of Statistics: Data Mining and Computational Statistics*. Elsevier, Amsterdam.

Urbanek, S., & Dasu T. (2007). A statistical framework for mining data streams. *Tutorial presentation, SIAM International Conference on Data Mining.*

Wong, P. C., Foote, H., Adams, D. Cowley, W., & Thomas, J. (2003). Dynamic visualization of transient data streams. In *Proc. of INFOVIS, 2003.* pp. 97-104.

Zhu, Y. and Shasha, D. (2002) StatStream: Statistical monitoring of thousands of data streams in real time**.** In *VLDB 2002*, pages 358--369.

KEY TERMS

Bootstrap: A technique by which multiple samples are created from a single sample to compute error bounds for statistics computed from the original sample. Efron & Tibshirani (1993).

Chi-Square Test: A statistical test based on the Chi-square distribution to determine the statistical significance of a sample statistic.

Histogram: A histogram is a mapping that counts the number of observations that fall into various disjoint categories (known as bins).

Hypothesis Testing: A statistical framework for making decisions using relatively small samples of data. Bickel & Doksum (2001).

IDS: Intrusion detection system is software installed at critical points in a network to monitor the data packets that pass through for suspicious patterns.

Join Key (Match Key): An attribute or field in a database used to join or combine different database tables.

Kullback-Leibler Distance: A measure of the divergence between two probability distributions. Cover & Thomas (1991).

Partition: A method of dividing an attribute space into mutually exclusive classes that completely cover the space.

Quantiles: Values of a random variable that mark off certain probability cut-offs of the distribution. For example, the median is the 50% quantile of a distribution.

Sampling Distribution: The empirical distribution of a statistic computed from multiple samples drawn from the same populations under the same conditions.

Mining Data with Group Theoretical Means

Gabriele Kern-Isberner
University of Dortmund, Germany

INTRODUCTION

Knowledge discovery refers to the process of extracting new, interesting, and useful knowledge from data and presenting it in an intelligible way to the user. Roughly, knowledge discovery can be considered a three-step process: preprocessing data; data mining, in which the actual exploratory work is done; and interpreting the results to the user. Here, I focus on the data-mining step, assuming that a suitable set of data has been chosen properly.

The patterns that we search for in the data are plausible relationships, which agents may use to establish cognitive links for reasoning. Such plausible relationships can be expressed via association rules. Usually, the criteria to judge the relevance of such rules are either frequency based (Bayardo & Agrawal, 1999) or causality based (for Bayesian networks, see Spirtes, Glymour, & Scheines, 1993). Here, I will pursue a different approach that aims at extracting what can be regarded as structures of knowledge — relationships that may support the inductive reasoning of agents and whose relevance is founded on information theory. The method that I will sketch in this article takes numerical relationships found in data and interprets these relationships as structural ones, using mostly algebraic techniques to elaborate structural information.

BACKGROUND

Common sense and expert knowledge is most generally expressed by rules, connecting a precondition and a conclusion by an if-then construction. For example, you avoid puddles on sidewalks because you are aware of the fact that if you step into a puddle, then your feet might get wet; similarly, a physician would likely expect a patient showing the symptoms of fever, headache, and a sore throat to suffer from a flu, basing his diagnosis on the rule that if a patient has a fever, headache, and sore throat, then the ailment is a flu, equipped with a sufficiently high probability.

If-then rules are more formally denoted as conditionals. The crucial point with conditionals is that they carry generic knowledge that is applicable to different situations. This fact makes them most interesting objects in artificial intelligence, in a theoretical as well as in a practical respect. For instance, a sales assistant who has a general knowledge about the preferences of his or her customers can use this knowledge when consulting any new customer.

Typically, two central problems have to be solved in practical applications: First, where do the rules come from? How can they be extracted from statistical data? And second, how should rules be represented? How should conditional knowledge be propagated and combined for further inferences? Both of these problems can be dealt with separately, but it is most rewarding to combine them, that is, to discover rules that are most relevant with respect to some inductive inference formalism and to build up the best model from the discovered rules that can be used for queries.

MAIN THRUST

This article presents an approach to discover association rules that are most relevant with respect to the maximum entropy methods. Because entropy is related to information, this approach can be considered as aiming to find the most informative rules in data. The basic idea is to exploit numerical relationships that are observed by comparing (relative) frequencies, or ratios of frequencies, and so forth, as manifestations of interactions of underlying conditional knowledge.

My approach differs from usual knowledge discovery and data-mining methods in various respects:

- It explicitly takes the instrument of inductive inference into consideration.

- It is based on statistical information but not on probabilities close to 1; actually, it mostly uses only structural information obtained from the data.
- It is not based on observing conditional independencies (as for learning causal structures), but aims at learning relevant conditional dependencies in a nonheuristic way.
- As a further novelty, it does not compute single, isolated rules, but yields a set of rules by taking into account highly complex interactions of rules.
- Zero probabilities computed from data are interpreted as missing information, not as certain knowledge.

The resulting set of rules may serve as a basis for maximum entropy inference. Therefore, the method described in this article addresses minimality aspects, as in Padmanabhan and Tuzhilin (2000), and makes use of inference mechanisms, as in Cristofor and Simovici (2002). Different from most approaches, however, it exploits the inferential power of the maximum entropy methods in full consequence and in a structural, nonheuristic way.

Modelling Conditional Knowledge by Maximum Entropy (ME)

Suppose a set $R^* = \{(B1|A1)[x1], \ldots, (Bn|An)[xn]\}$ of probabilistic conditionals is given. For instance, R^* may describe the knowledge available to a physician when he has to make a diagnosis. Or R^* may express common sense knowledge, such as "Students are young with a probability of (about) 80%" and "Singles (i.e., unmarried people) are young with a probability of (about) 70%", the latter knowledge being formally expressed by $R^* = \{ (young|student)[0.8], (young|single)[0.7] \}$.

Usually, these rule bases represent incomplete knowledge, in that a lot of probability distributions are apt to represent them. So learning or inductively representing the rules, respectively, means to take them as a set of conditional constraints and to select a unique probability distribution as the best model that can be used for queries and further inferences. Paris (1994) investigates several inductive representation techniques in a probabilistic framework and proves that the principle of maximum entropy (ME-principle) yields the only method to represent incomplete knowledge in an unbiased way, satisfying a set of postulates describing sound common sense reasoning. The entropy H(P) of a probability distribution P is defined as

$$H(P) = - \Sigma_w P(w) \log P(w),$$

where the sum is taken over all possible worlds, w, and measures the amount of indeterminateness inherent to P. Applying the principle of maximum entropy, then, means to select the unique distribution $P^* = ME(R^*)$ that maximizes H(P) among all distributions P that satisfy the rules in R^*. In this way, the ME-method ensures that no further information is added, so the knowledge R^* is represented most faithfully.

Indeed, the ME-principle provides a most convenient and founded method to represent incomplete probabilistic knowledge (efficient implementations of ME-systems are described in Roedder & Kern-Isberner, 2003). In an ME-environment, the expert has to list only whatever relevant conditional probabilities he or she is aware of. Furthermore, ME-modelling preserves the generic nature of conditionals by minimizing the amount of information being added, as shown in Kern-Isberner (2001).

Nevertheless, modelling ME-rule bases has to be done carefully so as to ensure that *all* relevant dependencies are taken into account. This task can be difficult and troublesome. Usually, the modelling rules are based somehow on statistical data. So, a method to compute rule sets appropriate for ME-modelling from statistical data is urgently needed.

Structures of Knowledge

The most typical approach to discover interesting rules from data is to look for rules with a significantly high (conditional) probability and a concise antecedent (Bayardo & Agrawal, 1999; Agarwal, Aggarwal, & Prasad, 2000; Fayyad & Uthurusamy, 2002; Coenen, Goulbourne, & Leng, 2001). Basing relevance on frequencies, however, is sometimes unsatisfactory and inadequate, particularly in complex domains such as medicine. Further criteria to measure the interestingness of the rules or to exclude redundant rules have also been brought forth (Jaroszewicz & Simovici, 2001; Bastide, Pasquier, Taouil, Stumme, & Lakhal, 2000; Zaki, 2000). Some of these algorithms also make use of optimization criteria, which are based on entropy (Jaroszewicz & Simovici, 2002).

Mostly, the rules are considered as isolated pieces of knowledge; no interaction between rules can be taken into account. In order to obtain more structured information, one often searches for causal relationships by investigating conditional independencies and thus noninteractivity between sets of variables (Spirtes et al., 1993).

Although causality is undoubtedly most important for human understanding, the concept seems to be too rigid to represent human knowledge in an exhaustive way. For instance, a person suffering from a flu is certainly sick (P(sick | flu) = 1), and he or she often will complain about headaches (P(headache | flu) = 0.9). Then you have P(headache | flu) = P(headache | flu & sick), but you would surely expect that P(headache | not flu) is different from P(headache | not flu & sick)! Although the first equality suggests a conditional independence between sick and headache, due to the causal dependency between headache and flu, the second inequality shows this to be (of course) false. Furthermore, a physician might also state some conditional probability involving sickness and headache, so you obtain a complex network of rules. Each of these rules will be considered relevant by the expert, but none will be found when searching for conditional independencies! So what, exactly, are the *structures of knowledge* by which conditional dependencies (not independencies! See also Simovici, Cristofor, D., & Cristofor, L., 2000) manifest themselves in data?

To answer this question, the *theory of conditional structures* has been presented in Kern-Isberner (2000). Conditional structures are an algebraic means to make the effects of conditionals on possible worlds (i.e., possible combinations or situations) transparent, in that they reflect whether the corresponding world verifies the conditional or falsifies it, or whether the conditional cannot be applied to the world because the *if*-condition is not satisfied. Consider, for instance, the conditional "If you step in a puddle, then your feet might get wet." In a particular situation, the conditional is applicable (you actually step into a puddle) or not (you simply walk around it), and it can be found verified (you step in a puddle and indeed, your feet get wet) or falsified (you step in a puddle, but your feet remain dry because you are wearing rain boots).

This intuitive idea of considering a conditional as a three-valued event is generalized in Kern-Isberner (2000) to handle the simultaneous impacts of a set of conditionals by using algebraic symbols for positive and negative impact, respectively. Then for each world, a word of these symbols can be computed, which shows immediately how the conditionals interact on this world. The proper mathematical structure for building words are (semi)groups, and indeed, group theory provides the basis for connecting numerical to structural information in an elegant way. In short, a probability (or frequency) distribution is called *(conditionally) indifferent with respect to a set of conditionals R** iff its numerical information matches the structural information provided by conditional structures. In particular, each ME-distribution turns out to be indifferent with respect to a generating set of conditionals.

Data Mining and Group Theory — A Strange Connection?

The concept of conditional structures, however, is not only an algebraic means to judge well-behavedness with respect to conditional information. The link between numerical and structural information, which is provided by the concept of conditional indifference, can also be used in the other direction, that is, to derive structural information about the underlying conditional relationships from numerical information. More precisely, finding a set of rules with the ability to represent a given probability distribution *P* via ME-methods can be done by elaborating numerical relationships in *P*, interpreting them as manifestations of underlying conditional dependencies. The procedure to discover appropriate sets of rules is sketched in the following steps:

- Start with a set *B* of simple rules, the length of which is considered to be large enough to capture all relevant dependencies.
- Search for numerical relationships in *P* by investigating which products of probabilities match.
- Compute the corresponding conditional structures with respect to *B*, yielding equations of group elements.
- Solve these equations by forming appropriate factor groups.
- Building these factor groups corresponds to eliminating and joining the basic conditionals in *B* to make their information more concise, in accordance with the numerical structure of *P*. Actually, the antecedents of the conditionals in *B* are shortened so as to comply with the numerical relationships in *P*.

So the basic idea of this algorithm is to start with long rules and to shorten them in accordance with the probabilistic information provided by *P* without losing information.

Group theory actually provides an elegant framework, on the one hand, to disentangle highly complex conditional interactions in a systematic way, and on the other hand, to make operations on the conditionals computable, which is necessary to make information more concise.

How to Handle Sparse Knowledge

The frequency distributions calculated from data are mostly not positive — just to the contrary, they would be sparse, full of zeros, with only scattered clusters of nonzero probabilities. This overload of zeros is also a problem with respect to knowledge representation, because a zero in such a frequency distribution often merely means that such a combination has not been recorded. The strict probabilistic interpretation of zero probabilities, however, is that such a combination does not exist, which does not seem to be adequate.

The method sketched in the preceding section is also able to deal with that problem in a particularly adequate way: The zero values in frequency distributions are taken to be unknown but equal probabilities, and this fact can be exploited by the algorithm. So they actually help to start with a tractable set B of rules right from the beginning (see also Kern-Isberner & Fisseler, 2004).

In summary, zeros occurring in the frequency distribution computed from data are considered as missing information, and in my algorithm, they are treated as non-knowledge without structure.

FUTURE TRENDS

Although by and large, the domain of knowledge discovery and data mining is dominated by statistical techniques and the problem of how to manage vast amounts of data, the increasing need for and popularity of human-machine interactions will make it necessary to search for more structural knowledge in data that can be used to support (humanlike) reasoning processes. The method described in this article offers an approach to realize this aim. The conditional relationships that my algorithm reveals can be considered as kind of cognitive links of an ideal agent, and the ME-technology takes the task of inductive reasoning to make use of this knowledge. Combined with clustering techniques in large databases, for example, it may turn out a useful method to discover relationships that go far beyond the results provided by other, more standard data-mining techniques.

CONCLUSION

In this article, I have developed a new method for discovering conditional dependencies from data. This method is based on information-theoretical concepts and group-theoretical techniques, considering knowledge discovery as an operation inverse to inductive knowledge representation. By investigating relationships between the numerical values of a probability distribution *P,* the effects of conditionals are analyzed and isolated, and conditionals are joined suitably so as to fit the knowledge structures inherent to *P.*

REFERENCES

Agarwal, R. C., Aggarwal, C. C., & Prasad, V. V. V. (2000). Depth first generation of long patterns. *Proceedings of the Sixth ACM-SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 108-118).

Bastide, Y., Pasquier, N., Taouil, R., Stumme, G. & Lakhal, L. (2000). Mining minimal non-redundant association rules using frequent closed itemsets. *Proceedings of the First International Conference on Computational Logic* (pp. 972-986).

Bayardo, R. J., & Agrawal, R. (1999). Mining the most interesting rules. *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.*

Coenen, F., Goulbourne, G., & Leng, P. H. (2001). Computing association rules using partial totals. *Proceedings of the Fifth European Conference on Principles and Practice of Knowledge Discovery in Databases* (pp. 54-66).

Cristofor, L., & Simovici, D. (2002). Generating an informative cover for association rules. *Proceedings*

of the IEEE International Conference on Data Mining (pp. 597-600).

Fayyad, U., & Uthurusamy, R. (2002). Evolving data mining into solutions for insights. *Communications of the ACM, 45*(8), 28-61.

Jaroszewicz, S., & Simovici, D. A. (2001). A general measure of rule interestingness. *Proceedings of the Fifth European Conference on Principles and Practice of Knowledge Discovery in Databases* (pp. 253-265).

Jaroszewicz, S., & Simovici, D. A. (2002). Pruning redundant association rules using maximum entropy principle. *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining.*

Kern-Isberner, G. (2000). Solving the inverse representation problem. *Proceedings of the 14th European Conference on Artificial Intelligence* (pp. 581-585).

Kern-Isberner, G. (2001). Conditionals in nonmonotonic reasoning and belief revision. *Lecture Notes in Artificial Intelligence*.

Kern-Isberner, G., & Fisseler, J. (2004). Knowledge discovery by reversing inductive knowledge representation. *Proceedings of the Ninth International Conference on the Principles of Knowledge Representation and Reasoning*.

Padmanabhan, B., & Tuzhilin, A. (2000). Small is beautiful: Discovering the minimal set of unexpected patterns. *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 54-63).

Paris, J. B. (1994). *The uncertain reasoner's companion: A mathematical perspective*. Cambridge University Press.

Roedder, W., & Kern-Isberner, G. (2003). From information to probability: An axiomatic approach. *International Journal of Intelligent Systems, 18*(4), 383-403.

Simovici, D.A., Cristofor, D., & Cristofor, L. (2000). *Mining for purity dependencies in databases* (Tech. Rep. No. 00-2). Boston: University of Massachusetts.

Spirtes, P., Glymour, C., & Scheines, R.. (1993). *Causation, prediction and search*. *Lecture Notes in Statistics, 81*.

Zaki, M. J. (2000). Generating non-redundant association rules. *Proceedings of the Sixth ACM-SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 34-43).

KEY TERMS

Conditional: The formal algebraic term for a rule that need not be strict, but also can be based on plausibility, probability, and so forth.

Conditional Independence: A generalization of plain statistical independence that allows you to take a context into account. Conditional independence is often associated with causal effects.

Conditional Structure: An algebraic expression that makes the effects of conditionals on possible worlds transparent and computable.

Entropy: Measures the indeterminateness inherent to a probability distribution and is dual to information.

Possible World: Corresponds to the statistical notion of an elementary event. Probabilities over possible worlds, however, have a more epistemic, subjective meaning, in that they are assumed to reflect an agent's knowledge.

Principle of Maximum Entropy: A method to complete incomplete probabilistic knowledge by minimizing the amount of information added.

Probabilistic Conditional: A conditional that is assigned a probability. To match the notation of conditional probabilities, a probabilistic conditional is written as (B|A)[x] with the meaning "If A holds, then B holds with probability x."

Mining Email Data

Steffen Bickel
Humboldt-Universität zu Berlin, Germany

Tobias Scheffer
Humboldt-Universität zu Berlin, Germany

INTRODUCTION

E-mail has become one of the most important communication media for business and private purposes. Large amounts of past e-mail records reside on corporate servers and desktop clients. There is a huge potential for mining this data. E-mail filing and spam filtering are well-established e-mail mining tasks. E-mail filing addresses the assignment of incoming e-mails to predefined categories to support selective reading and organize large e-mail collections. First research on e-mail filing was conducted by Green and Edwards (1996) and Cohen (1996). Pantel and Lin (1998) and Sahami, Dumais, Heckerman, and Horvitz (1998) first published work on spam filtering. Here, the goal is to filter unsolicited messages. Recent research on e-mail mining addresses automatic e-mail answering (Bickel & Scheffer, 2004) and mining social networks from e-mail logs (Tyler, Wilkinson, & Huberman, 2004).

In Section *Background* we will categorize common e-mail mining tasks according to their objective, and give an overview of the research literature. Our *Main Thrust* Section addresses e-mail mining with the objective of supporting the message creation process. Finally, we discuss *Future Trends* and conclude.

BACKGROUND

There are two objectives for mining e-mail data: supporting communication and discovering hidden properties of communication networks.

Support of Communication

The problems of filing e-mails and filtering spam are text classification problems. Text classification is a well studied research area; a wide range of different methods is available. Most of the common text classification algorithms have been applied to the problem of e-mail classification and their performance has been compared in several studies. Because publishing an e-mail data set involves disclosure of private e-mails, there are only a small number of standard e-mail classification data sets. Since there is no study that compares large numbers of data sets, different classifiers and different types of extracted features, it is difficult to judge which text classifier performs best specifically for e-mail classification.

Against this background we try to draw some conclusions on the question which is the best text classifier for e-mail. Cohen (1996) applies rule induction to the e-mail classification problem and Provost (1999) finds that Naïve Bayes outperforms rule induction for e-mail filing. Naïve Bayes classifiers are widely used for e-mail classification because of their simple implementation and low computation time (Pantel & Lin, 1998; Rennie, 2000; Sahami, Dumais, Heckerman, & Horvitz, 1998). Joachims (1997, 1998) shows that Support Vector Machines (SVMs) are superior to the Rocchio classifier and Naïve Bayes for many text classification problems. Drucker, Wu, and Vapnik (1999) compares SVM with boosting on decision trees. SVM and boosting show similar performance but SVM proves to be much faster and has a preferable distribution of errors.

The performance of an e-mail classifier is dependent on the extraction of appropriate features. Joachims (1998) shows that applying feature selection for text classification with SVM does not improve performance. Hence, using SVM one can bypass the expensive feature selection process and simply include all available features. Features that are typically used for e-mail classification include all tokens in the e-mail body and header in bag-of-words representation using TF- or TFIDF-weighting. HTML tags and single URL elements also provide useful information (Graham, 2003).

Boykin and Roychowdhury (2004) propose a spam filtering method that is not based on text classification

but on graph properties of message sub-graphs. All addresses that appear in the headers of the inbound mails are graph nodes; an edge is added between all pairs of addresses that jointly appear in at least one header. The resulting sub-graphs exhibit graph properties that differ significantly for spam and non-spam sub-graphs. Based on this finding "black-" and "whitelists" can be constructed for spam and non-spam addresses. While this idea is appealing, it should be noted that the approach is not immediately practical since most headers of spam e-mails do not contain other spam recipients' addresses, and most senders' addresses are used only once.

Additionally, the "*semantic e-mail"* approach (McDowell, Etzioni, Halevy, & Levy, 2004) aims at supporting communication by allowing automatic e-mail processing and facilitating e-mail mining; it is the equivalent of *semantic web* for e-mail. The goal is to make e-mails human- and machine-understandable with a standardized set of e-mail processes. Each e-mail has to follow a standardized process definition that includes specific process relevant information. An example for a *semantic e-mail* process is meeting coordination. Here, the individual process tasks (corresponding to single e-mails) are issuing invitations and collecting responses. In order to work, *semantic e-mail* would require a global agreement on standardized semantic processes, special e-mail clients and training for all users. Additional mining tasks for support of communication are automatic e-mail answering and sentence completion. They are described in Section *Main Thrust*.

Discovering Hidden Properties of Communication Networks

E-mail communication patterns reveal much information about hidden social relationships within organizations. Conclusions about informal communities and informal leadership can be drawn from e-mail graphs. Differences between informal and formal structures in business organizations can provide clues for improvement of formal structures which may lead to enhanced productivity. In the case of terrorist networks, the identification of communities and potential leaders is obviously helpful as well. Additional potential applications lie in marketing, where companies – especially communication providers – can target communities as a whole.

In social science, it is common practice for studies on electronic communication within organizations to derive the network structure by means of personal interviews or surveys (Garton Garton, Haythornthwaite, & Wellman, 1997; Hinds & Kiesler, 1995). For large organizations, this is not feasible. Building communication graphs from e-mail logs is a very simple and accurate alternative provided that the data is available. Tyler, Wilkinson, and Huberman (2004) derive a network structure from e-mail logs and apply a divisive clustering algorithm that decomposes the graph into communities. Tyler, Wilkinson, and Huberman verify the resulting communities by interviewing the communication participants; they find that the derived communities correspond to informal communities.

Tyler et al. also apply a force-directed spring algorithm (Fruchterman & Rheingold, 1991) to identify leadership hierarchies. They find that with increasing distance of vertices from the "spring" (center) there is a tendency of decreasing real hierarchy depth.

E-mail graphs can also be used for controlling virus attacks. Ebel, Mielsch, and Bornholdt (2002) show that vertex degrees of e-mail graphs are governed by power laws. By equipping the small number of highly connected nodes with anti-virus software the spreading of viruses can be prevented easily.

MAIN THRUST

In the last section we categorized e-mail mining tasks regarding their objective and gave a short explanation on the single tasks. We will now focus on the ones that we consider to be most interesting and potentially most beneficial for users and describe them in greater detail. These tasks aim at supporting the message creation process. Many e-mail management systems allow the definition of message templates that simplify the message creation for recurring topics. This is a first step towards supporting the message creation process, but past e-mails that are available for mining are disregarded. We describe two approaches for supporting the message creation process by mining historic data: mining question-answer pairs and mining sentences.

Mining Question-Answer Pairs

We consider the problem of learning to answer incoming e-mails from records of past communication.

We focus on environments in which large amounts of similar answers to frequently asked questions are sent – such as call centers or customer support departments. In these environments, it is possible to *manually* identify equivalence classes of answers in the records of *outbound* communication. Each class then corresponds to a set of semantically equivalent answers sent in the past; it depends strongly on the application context which fraction of the outbound communication falls into such classes. Mapping *inbound* messages to one of the equivalence classes of answers is now a multi-class text classification problem that can be solved with text classifiers.

This procedure requires a user to manually group previously sent answers into equivalence classes which can then serve as class labels for training a classifier. This substantial manual labeling effort reduces the benefit of the approach. Even though it can be reduced by employing semi-supervised learning (Nigam, McCallum, Thrun, & Mitchell, 2000; Scheffer, 2004), it would still be much preferable to learn from only the available data: stored inbound and outbound messages. Bickel and Scheffer (2004) discuss an algorithm that learns to answer questions from only the available data and does not require additional manual labeling. The key idea is to replace the manual assignment of outbound messages to equivalence classes by a clustering step.

The algorithms for training (learning from message pairs) and answering a new question are shown in Table 1. In the training phase, a clustering algorithm identifies groups of similar outbound messages. Each cluster then serves as class label; the corresponding questions which have been answered by a member of the cluster are used as training examples for a multi-class text classifier. The medoid of each cluster (the outbound message closest to the center) is used as an answer template. The classifier maps a newly incoming question to one of the clusters; this cluster's medoid is then proposed as answer to the question. Depending on the user interface, high confidence messages might be answered automatically, or an answer is proposed which the user may then accept, modify, or reject (Scheffer, 2004).

The approach can be extended in many ways. Multiple topics in a question can be identified to mix different corresponding answer templates and generate

Table 1. Algorithms for learning from message pairs and answering new questions.

Learning from message pairs.

Input: Message pairs, variance threshold σ^2, pruning parameter π.

1. Recursively cluster answers of message pairs with bisecting partitioning cluster algorithm, end recursion when cluster variance lies below σ^2.
2. Prune all clusters with less than π elements. Combine all pruned clusters into one "miscellaneous" cluster. Let n be the number of resulting clusters.
3. For all n clusters
    a. Construct an answer template by choosing the answer that is most similar to the centroid of this cluster in vector space representation and remove salutation line.
    b. Let the inbound mails that have been answered by a mail in the current cluster be the positive training examples for this answer class.
2. Train SVM classifier that classifies an inbound message into one of the n answer classes or the "miscellaneous" class from these training examples. Return this classifier.

Answering new questions.

Input: New question message, message answering hypothesis, confidence threshold θ.

1. Classify new message into one of the n answer classes and remember SVM decision function value.
2. If confidence exceeds the confidence threshold, propose the answer template that corresponds to the classification result. Perform instantiation operations that typically include formulating a salutation line.

a multi-topic answer. Question specific information can be extracted in an additional information extraction step and automatically inserted into answer templates. In this extraction step also customer identifications can be extracted and used for a database lookup that provides customer and order specific information for generating more customized answers.

Bickel and Scheffer (2004) analyze the relationship of answer classes regarding the separability of the corresponding questions using e-mails sent by the service department of an online shop. By analyzing this relationship one can draw conclusions about the amount of additional information that is needed for answering specific types of questions. This information can be visualized in an *inseparability graph*, where each class of equivalent answers is represented by a vertex, and an edge is drawn when a classifier that discriminates between these classes achieves only a low AUC performance (the AUC performance is the probability that, when a positive and a negative example are drawn at random, a discriminator assigns a higher value to the positive than to the negative one). Typical examples of inseparable answers are "your order has been shipped this morning" and "your order will be shipped tomorrow". Intuitively, it is not possible to predict which of these answers a service employee will send, based on only the question "when will I receive my shipment?"

Mining Sentences

The message creation process can also be supported on a sentence level. Given an incomplete sentence, the task of *sentence completion* is to propose parts or the total rest of the current sentence, based on an application specific document collection. A sentence completion user interface can, for instance, display a proposed completion in a "micro window" and insert the proposed text when the user presses the "tab" key.

The sentence completion problem poses new challenges for data mining and information retrieval, including the problem of finding sentences whose initial fragment is similar to a given fragment in a very large text corpus. To this end, Grabski and Scheffer (2004) provide a retrieval algorithm that uses a special inverted indexing structure to find the sentence whose initial fragment is most similar to a given fragment, where similarity is defined in terms of the greatest cosine similarity of the TFIDF vectors. In addition, they study an approach that compresses the data further by identifying clusters of the most frequently used similar sets of sentences. In order to evaluate the accuracy of sentence completion algorithms, Grabski and Scheffer (2004) measure how frequently the algorithm, when given a sentence fragment drawn from a corpus, provides a prediction with confidence above θ, and how frequently this prediction is semantically equivalent to the actual sentence in the corpus. They find that for the sentence mining problem higher precision and recall values can be obtained than for the problem of mining question answer pairs; depending on the threshold θ and the fragment length, precision values of between 80% and 100% and recall values of about 40% can be observed.

FUTURE TRENDS

Spam filtering and e-mail filing based on message text can be reduced to the well studied problem of text classification. The challenges that e-mail classification faces today concern technical aspects, the extraction of spam-specific features from e-mails, and an arms race between spam filters and spam senders adapting to known filters. By comparison, research in the area of automatic e-mail answering and sentence completion is in an earlier stage; we see a substantial potential for algorithmic improvements to the existing methods. The technical integration of these approaches into existing e-mail clients or call-center automation software provides an additional challenge. Some of these technical challenges have to be addressed before mining algorithms that aim at supporting communication can be evaluated under realistic conditions.

Construction of social network graphs from e-mail logs is much easier than by surveys and there is a huge interest in mining social networks – see, for instance, the DARPA program on Evidence Extraction and Link Discovery (EELD). While social networks have been studied intensely in the social sciences and in physics, we see a considerable potential for new and better mining algorithms for social networks that computer scientists can contribute.

CONCLUSION

Some methods that can form the basis for effective spam filtering have reached maturity (text classification),

additional foundations are being worked on (social network analysis). Today, technical challenges dominate the development of spam filters. The development of methods that support and automate communication processes is a research topic and first solutions to some of the problems involved have been studied. Mining social networks from e-mail logs is a new challenge; research on this topic in computer science is in an early stage.

ACKNOWLEDGMENT

The authors are supported by the German Science Foundation DFG under grant SCHE540/10-1. We would like to thank the anonymous reviewers.

REFERENCES

Bickel, S., & Scheffer, T. (2004). Learning from message pairs for automatic email answering. *Proceedings of the European Conference on Machine Learning*.

Boykin, P., & Roychowdhury, V. (2004). *Personal e-mail networks: An effective anti-spam tool.* Preprint, arXiv id 0402143.

Cohen, W. (1996). Learning rules that classify e-mail. *Proceedings of the IEEE Spring Symposium on Machine learning for Information Access*, Palo Alto, California, USA.

Drucker, H., Wu, D., & Vapnik, V. (1999). Support vector machines for spam categorization. *IEEE Transactions on Neural Networks, 10*(5), 1048-1055.

Ebel, H., Mielsch, L., & Bornholdt, S. (2002). Scale-free topology of e-mail networks. *Physical Review*, E 66.

Fruchterman, T. M., & Rheingold, E. M. (1991). Force-directed placement. *Software Experience and Practice, 21*(11).

Garton, L., Haythornthwaite, C., & Wellman, B. (1997). Studying online social networks. *Journal of Computer-Mediated Communication, 3*(1).

Grabski, K., & Scheffer, T. (2004). Sentence completion. *Proceedings of the SIGIR International Conference on Information Retrieval*, Sheffield, UK.

Graham, P. (2003). Better Bayesian filtering. *Proceedings of the First Annual Spam Conference,* MIT. Retrieved from *http://www.paulgraham.com/better.html*

Green, C., & Edwards, P. (1996). Using machine learning to enhance software tools for internet information management. *Proceedings of the AAAI Workshop on Internet Information Management*.

Hinds, P., & Kiesler, S. (1995). Communication across boundaries: Work, structure, and use of communication technologies in a large organization. *Organization Science, 6*(4), 373-393.

Joachims, T. (1997). A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. *Proceedings of the International Conference on Machine Learning.*

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. *Proceedings of the European Conference on Machine Learning*.

McDowell, L., Etzioni, O., Halevy, A., & Levy, H. (2004). Semantic e-mail. *Proceedings of the WWW Conference*.

Nigam, K., McCallum, A., Thrun, S., & Mitchell, T. (2000). Text classification from labeled and unlabeled documents using EM. *Machine Learning, 39*(2/3).

Pantel, P., & Lin, D. (1998) Spamcop: a spam classification and organization program. *Proceedings of the AAAI Workshop on Learning for Text Categorization*.

Provost, J. (1999). *Naïve Bayes vs. rule-learning in classification of e-mail.* Technical Report AI-TR-99-284, University of Texas at Austin.

Rennie, J. (2000). iFILE: An application of machine learning to e-mail filtering. *Proceedings of the SIGKDD Text Mining Workshop*.

Sahami, M., Dumais, S., Heckerman, D., & Horvitz, E. (1998). A Bayesian approach to filtering junk e-mail. *Proceedings of AAAI Workshop on Learning for Text Categorization*.

Scheffer, T. (2004). E-mail answering assistance by semi-supervised text classification. *Intelligent Data Analysis, 8*(5).

Tyler, J. R., Wilkinson, D. M., & Huberman, B. A. (2003). E-mail as spectroscopy: Automated discovery

of community structure within organizations. *Proceedings of the International Conference on Communities and Technologies* (pp. 81-95). Kluwer Academic Publishers.

KEY TERMS

Community: A group of people having mutual relationships among themselves or having common interests. Clusters in social network graphs are interpreted as communities.

Mining E-Mails: The application of analytical methods and tools to e-mail data for a) support of communication by filing e-mails into folders, filtering spam, answering e-mails automatically, or proposing completions to sentence fragments, b) discovery of hidden properties of communication networks by e-mail graph analysis.

Mining Question-Answer Pairs: Analytical method for automatically answering question e-mails using knowledge that is discovered in question-answer pairs of past e-mail communication.

Mining Sentences: Analytical method for interactively completing incomplete sentences using knowledge that is discovered in a document collection.

Semantic E-Mail: E-mail framework in which the semantics of e-mails is understandable by both, human and machine. A standardized definition of semantic e-mail processes is required.

Spam E-Mail: Unsolicited and unwanted bulk e-mail. Identifying spam e-mail is a text classification task.

Text Classification: The task of assigning documents expressed in natural language to one or more categories (classes) of a predefined set.

TFIDF: Weighting scheme for document and query representation in the vector space model. Each dimension represents a term, its value is the product the frequency of the term in a document (TF) and the inverse document frequency (IDF) of the term. The inverse document frequency of a term is the logarithmic proportion of documents in which the term occurs. The TFIDF scheme assigns a high weight to terms which occur frequently in the focused document, but are infrequent in average documents.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 768-772, copyright 2005 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

Mining Generalized Association Rules in an Evolving Environment

Wen-Yang Lin
National University of Kaohsiung, Taiwan

Ming-Cheng Tseng
Institute of Information Engineering, Taiwan

INTRODUCTION

The mining of Generalized Association Rules (GARs) from a large transactional database in the presence of item taxonomy has been recognized as an important model for data mining. Most previous studies on mining generalized association rules, however, were conducted on the assumption of a static environment, i.e., static data source and static item taxonomy, disregarding the fact that the taxonomy might be updated as new transactions are added into the database over time, and as such, the analysts may have to continuously change the support and confidence constraints, or to adjust the taxonomies from different viewpoints to discover more informative rules. In this chapter, we consider the problem of mining generalized association rules in such a dynamic environment. We survey different strategies incorporating state-of-the-art techniques for dealing with this problem and investigate how to efficiently update the discovered association rules when there are transaction updates to the database along with item taxonomy evolution and refinement of support constraint.

BACKGROUND

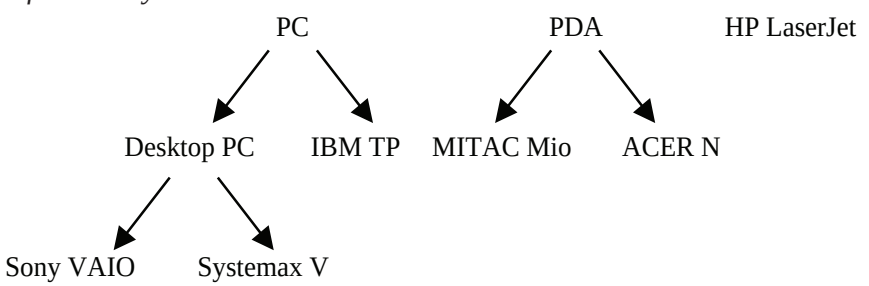
An association rule is an expression of the form $X \Rightarrow Y$, where X and Y are sets of items. Such a rule reveals that transactions in the database containing items in X tend to also contain items in Y, and the probability, measured as the fraction of transactions containing X that also contain Y, is called the *confidence* of the rule. The *support* of the rule is the fraction of the transactions that contain all items in both X and Y. The problem of mining association rules is to discover all association rules that satisfy support and confidence constraints.

In many applications, there are explicit or implicit taxonomies over the items, so it may be more useful to find associations at different taxonomic levels than only at the primitive concept level. For example, consider the taxonomy of items in Figure 1. It is likely that the association rule,

Systemax V $\Rightarrow$ HP LaserJet ($sup \geq 20\%$, $conf = 100\%$) does not hold when the minimum support is set to 25%, but the following association rule may be valid,

Desktop PC $\Rightarrow$ HP LaserJet

Figure 1. Example of taxonomy



This kind of association rule with taxonomy is also called the *generalized association rule* (Srikant & Agrawal, 1995) or the *multi-level association rule* (Han & Fu, 1995). The work in Srikant and Agrawal (1995) aimed at finding associations among items at any level of the taxonomy, whereas the objective in Han and Fu (1995) was to discover associations level-by-level in a fixed hierarchy, i.e., only associations among items on the same level were examined progressively from the top level to the bottom.

Mining GARs with Transaction Update

In the real world, however, data mining practitioners are usually confronted with a dynamic environment. The first challenge comes from the fact that the source database is not static. New transactions are continually added into the database over time, outdated transactions are occasionally or periodically purged from the repository, and some transactions might be modified, thus leading to the essence of updating discovered association rules when transaction updates occur in the database over time. Cheung et al. (1996a) first addressed this problem and proposed an algorithm called FUP (Fast UPdate). They further extended the model to incorporate the situations of deletion and modification (Cheung et al., 1997). Subsequently, a number of techniques have been proposed to improve the efficiency of incremental mining algorithms, such as *negative border* (Sarda & Srinivas, 1998; Thomas et al., 1997), *dynamic counting* (Ng & Lam, 2000), *pre-large itemsets* (Hong et al., 2001), *sliding window filter* (Lee et al., 2005), *support prediction* (Guirguis et al., 2006), and *FP-like tree structure* (Ezeife & Su, 2002; Leung et al., 2007), although all of these were confined to mining associations among primitive items.

The maintenance issue for generalized association rules was also first studied by Cheung et al. (1996b), who proposed an extension of their FUP algorithm, called MLUp, to accomplish the task. Hong et al. (2004) then extended Han and Fu's approach (Han & Fu, 1995) by introducing the concept of pre-large itemsets (Hong et al., 2000) to postpone the original database rescanning until a number of records have been modified. In (Tseng & Lin, 2004), Tseng and Lin extended the problem to incorporate non-uniform minimum support.

Mining GARs with Interestingness Refinement

The second challenge arises from users' perception of information needs. Faced with an unknown and large volume of data collected under a highly competitive environment, analysts generally lack of knowledge about the application domains and so have to change their viewpoints continuously, in an interactive way to find informative rules. In the context of association mining, the user's viewpoint, in its simplest form, is specified through the rule's thresholds, e.g., minimum support and minimum confidence. In the past decade, various strategies have been proposed to realize the interactive (or online) association mining, including *precomputation* (Han, 1998; Aggarwal & Yu, 2001; Czejdo et al., 2002; Duan et al., 2006; Liu et al., 2007), *caching* (Nag et al., 1999), and *incremental update* (Hidber, 1999; Liu & Yin, 2001; Ma et al., 2002; Deng et al., 2005).

The general idea for precomputation strategy is to precompute all frequent itemsets relative to a presetting support threshold, and once the specified support threshold is larger than the presetting value, the qualified association rules can be immediately generated without the burden of an expensive phase for itemset generation. With a similar philosophy, the caching strategy tries to eliminate the cost spent on frequent itemsets computation by temporarily storing previously discovered frequent itemsets (may be accompanied with some infrequent itemsets) that are beneficial to subsequent association queries. The effectiveness of this strategy relies primarily on a successful design of the cache replacement algorithm. The third strategy can be regarded as a compensation for the first two strategies. During the course of a sequence of remining trials with varied support thresholds, the incremental update strategy endeavors to utilize the discovered frequent itemsets in previous trial to reduce the cost for subsequent re-execution.

Mining GARs with Taxonomy Evolution

The third challenge comes to the evolution of item taxonomy. As a representation of the classification relationship imposed on items, a taxonomy must evolve to reflect what has occurred to the domain applications. For example, items corresponding to new products must be added into the taxonomy, and their insertion

would further introduce new classifications if they are of new types. On the other hand, outdated items along with their classifications will also be abandoned if they are no longer produced. All of these changes would reshape the taxonomy and in turn invalidate previously discovered generalized association rules and/or introduce new rules.

In addition to the evolution of taxonomies due to application background change, analysts might also like to adjust the taxonomies from different viewpoints to discover more informative rules (Han & Fu, 1994). Consider Figure 1 for example. The analysts would like to add an additional level, say "Product Vendor", into the taxonomy to discover rules that exploit associations involving such a concept.

Compared with the substantial amount of research on the first two challenges there is a lack of effort devoted to the problem of mining GARs with taxonomy evolution. Lin and Tseng (Lin & Tseng, 2004; Tseng et al., 2005b; Tseng et al., 2006) have considered this problem and some of its extensions, and proposed efficient Apriori-like algorithms.

MAIN FOCUS

System Framework for Mining Evolving GARs

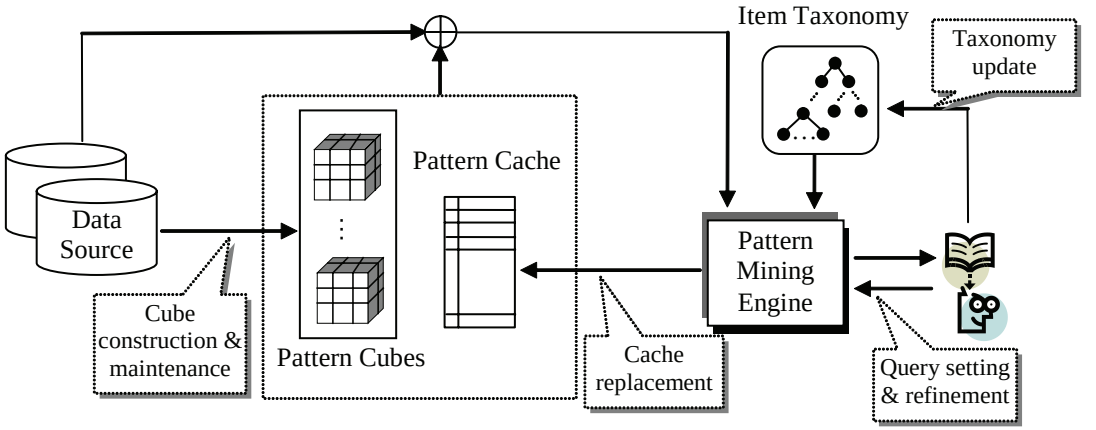
In light of the various considerations addressed in the previous section for mining GARs in an evolving environment, we conjecture that the best way is to integrate various techniques developed so far for dealing with only one or two aspects of the problem. Towards this goal, we propose a general framework, as depicted in Figure 2, which can accommodate all the state-of-the-art techniques to accomplish interactive mining of GARs in an evolving environment.

Essential to this framework is an efficient frequent pattern mining engine that can handle the most complicated situation when new transactions are added to the source data, along with the taxonomy being updated by the system administrator or the user, and as such the user should refine the mining query with a new support threshold. In what follows, we describe our proposed Apriori-based incremental mining algorithm.

Incremental Algorithm for Mining Generalized Frequent Itemsets

Consider the situation when new transactions, denoted as *db*, are added to a database *DB* and the taxonomy *T* is changed into a new one *T**. Let *ED** and *ed** denote the extended version of the original database *DB* and incremental database *db*, respectively, by adding the generalized items in *T** to each transaction. Further, let *UE** be the updated extended database containing *ED** and *ed**, i.e., *UE** = *ED** + *ed**. To update the set of frequent itemsets in *ED*, L^{ED}, when new transactions *db* are added to *DB*, *T* is changed into *T**, and the old minimum support (ms_{old}) is changed into the new one (ms_{new}), this problem is equivalent to finding the set of frequent itemsets in *UE** with respect to ms_{new}, denoted as L^{UE*}. Based on an Apriori-like framework, our algorithm employs a bottom-up search strategy, first discovering the set of all frequent 1-itemsets, and

Figure 2. General framework for GAR mining



proceeding towards the set of frequent itemsets with maximal cardinality *m*. Each pass for mining the frequent *k*-itemsets, for $1 \leq k \leq m$, involves the following main steps:

1. Generate candidate *k*-itemsets C_k.
2. Differentiate in C_k the affected itemsets (C_k^+) from the unaffected ones (C_k^-). Here the term affected itemsets refer to those itemsets whose support would be changed with respect to the taxonomy evolution.
3. Scan the incremental database *db* with the new taxonomy *T** to find frequent itemsets L_k^{ed*} with respect to the new support threshold ms_{new}.
4. Incorporate L_k^{ED}, L_k^{ed*}, C_k^+, C_k^-, ms_{old} and ms_{new} to determine whether a candidate itemset is frequent or not in the resulting database *UE**. Cases for the inference of frequent itemsets are shown in Table 1. Take case 5 for example: If *A* is an unaffected infrequent itemset in *ED* but is frequent in *ed**, then no matter whether $ms_{new} \leq ms_{old}$ or $ms_{new} > ms_{old}$, it is an undetermined itemset in *UE**.
5. Scan *DB* with *T**, i.e., *ED**, to count the supports of itemsets that are undetermined in Step 4.

Empirical Evaluation

To examine the performance of our algorithm, we conduct experiments to compare its performance with that of applying two leading generalized association mining algorithms, Cumulate and Stratify (Srikant & Agrawal, 1995), to the whole updated database with a new taxonomy under new support threshold. All experiments were performed on an Intel Pentium-IV 2.80GHz with 2GB RAM, running on Windows 2000. A synthetic dataset, generated by the IBM data generator (Agrawal & Srikant, 1994), with artifact taxonomy consisting of 5 groups of hierarchies with depth of 3 and average fanout of 5 is used in the experiments. The evaluations were examined from three aspects: the effect of varying minimum supports, that of varying incremental size and that of the degree of taxonomy evolution. All results showed that our algorithm is significantly faster than running Cumulate or Stratify from scratch.

FUTURE TRENDS

Since many current applications generate large volumes of data in a continuous and automated way, the problem of data stream mining has become an emerging issue (Babcock et al., 2002; Golab & Ozsu, 2003; Jin & Agrawal, 2007). The fast-changing and potentially infinite characteristics of stream data, however, make the problem of mining stream data very different from that of mining traditional transaction data. The most important difference is that "you can only look once", which implies that mining algorithms requiring more than a single pass of dataset scan are not applicable. In addition, memory limitation is always a significant concern. To the best of our knowledge, the taxonomy information and the effect of taxonomy evolution mining association rules over data streams has not yet been addressed in the literature.

Another promising avenue for future studies is the incorporation of domain knowledge other than tax-

Table 1. Cases for frequent candidate inference under incremental database update, minimum support refinement and taxonomy evolution

Conditions				Result	
$T \rightarrow T^*$	L^{ED} (ms_{old})	L^{ed*} (ms_{new})	ms_{new} vs ms_{old}	UE^* (ms_{new})	Case
unaffected	$\in$	$\in$	$\leq$	freq.	1
			$>$	undetd.	3
		$\notin$	$\leq, >$	undetd.	4
	$\notin$	$\in$	$\leq, >$	undetd.	5
		$\notin$	$<$	undetd.	6
			$\geq$	infreq.	2
affected	$\in, \notin$	$\in, \notin$	$\leq, >$	undetd.	7

onomy into the processing of association discovery. According to Storey (1993), the semantic relationships imposed on items in the real world can be classified into several categories, including possession, inclusion, attachment, attribution, antonyms, synonyms and case. Mining association rules incorporating such complex item relationships would be an interesting but very challenging issue.

CONCLUSION

In this chapter, we have investigated the problem of mining generalized association rules in an evolving environment and have surveyed state-of-the-art techniques for solving various aspects, including transaction update, threshold refinement, and taxonomy evolution of this problem. A general framework that can incorporate all of these techniques towards the interactive discovery of generalized association rules has been proposed. Of particular importance to the success of this system is an efficient algorithm that can incrementally update discovered frequent itemsets with respect to any kind of evolving aspects. Empirical studies have shown that the proposed algorithm is significantly faster than the simple strategy of re-mining frequent itemsets from scratch with the fastest algorithms.

REFERENCES

Aggarwal, C. C., & Yu, P. S. (2001). A new approach to online generation of association rules. *IEEE Transactions on Knowledge and Data Engineering*, 13(4), 527-540.

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules. In J. B. Bocca, M. Jarke, C. Zaniolo (Eds.), *Proceedings of the 20th International Conference on Very Large Data Bases* (pp. 487-499). Morgan Kaufmann.

Babcock, B., Babu, S., Datar, M., Motwani, R., & Widom, J. (2002). Models and issues in data stream systems. In L. Popa (Ed.), *Proceedings of the 21st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems* (pp. 1-16). ACM Press.

Cheung, D. W., Han, J., Ng, V. T., & Wong, C. Y. (1996a). Maintenance of discovered association rules in large databases: An incremental update technique. In S. Y. W. Su (Ed.), *Proceedings of the 12th International Conference on Data Engineering* (pp. 106-114). IEEE Computer Society.

Cheung, D. W., Ng, V. T., & Tam, B. W. (1996b). Maintenance of discovered knowledge: A case in multi-level association rules. In E. Simoudis, J. Han, U. M. Fayyad (Eds.), *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining* (pp. 307-310). AAAI Press.

Cheung, D. W., Lee, S. D., & Kao, B. (1997). A general incremental technique for maintaining discovered association rules. In R. W. Topor, K. Tanaka (Eds.), *Proceedings of the 5th International Conference on Database Systems for Advanced Applications* (pp. 185-194). Singapore: World Scientific Press.

Czejdo, B., Morzy, M., Wojciechowski, M., & Zakrzewicz, M. (2002). Materialized views in data mining. In A. Hameurlain, R. Cicchetti, R. Traunmüller (Eds.), *Proceedings of the 13th International Workshop on Database and Expert Systems Applications*, (pp. 827-834). IEEE Computer Society.

Deng, Z. H., Li, X., & Tang, S. W. (2005). An efficient approach for interactive mining of frequent itemsets. In W. Fan, Z. Wu, J. Yang (Eds.), *Proceedings of the 6th International Conference on Web-Age Information Management*, *Lecture Notes in Computer Science* 3739 (pp. 138-149). Springer.

Duan, Z., Cai, Z., & Lv, Y. (2006). Incremental maintenance of association rules based on multiple previously mined results. In X. Li, O. R. Zaiane, Z. Li (Eds.), *Proceedings of the 2nd International Conference on Advanced Data Mining and Applications*, *Lecture Notes in Computer Science* 4093 (pp. 72-79). Springer.

Ezeife, C. I., & Su, Y. (2002). Mining incremental association rules with generalized FP-tree. In R. Cohen, B. Spencer (Eds.), *Advances in Artificial Intelligence: 15th Conference of the Canadian Society for Computational Studies of Intelligence, Lecture Notes in Computer Science* 2338 (pp. 147-160). Springer.

Golab, L., & Ozsu, M. T. (2003). Issues in data stream management. *ACM SIGMOD Record*, 32(2), 5-14.

Guirguis, S., Ahmed, K. M., El Makky, N. M., & Hafez, A. M. (2006). Mining the future: Predicting itemsets' support of association rules mining. In *Workshop Pro-*

ceedings of the 6th IEEE International Conference on Data Mining (pp. 474-478). IEEE Computer Society.

Han, J. (1998). Toward on-line analytical mining in large databases. *ACM SIGMOD Record*, 27(1), 97-107.

Han, J., & Fu, Y. (1994). Dynamic generation and refinement of concept hierarchies for knowledge discovery in databases. In U. M. Fayyad, R. Uthurusamy (Eds.), *Proceedings of 1994 AAAI Workshop on Knowledge Discovery in Databases* (pp. 157-168). AAAI Press.

Han, J., & Fu, Y. (1995). Discovery of multiple-level association rules from large databases. In U. Dayal, P. M. D. Gray, S. Nishio (Eds.), *Proceedings of the 21st International Conference on Very Large Data Bases* (pp. 420-431). Morgan Kaufmann.

Hidber, C. (1999). Online association rule mining. *ACM SIGMOD Record*, 28(2), 145-156.

Hong, T. P., Wang, C. Y., & Tao, Y. H. (2001). A new incremental data mining algorithm using pre-large itemsets. *Intelligent Data Analysis*, 5(2), 111-129.

Hong, T. P., Huang, T. J., & Chang, C. S. (2004). Maintenance of multiple-level association rules for record modification. In *Proceedings of 2004 IEEE International Conference on Systems, Man & Cybernetics*, 3140-3145.

Jin, R., & Agrawal, G. (2007). Frequent pattern mining in data streams. In C. C. Aggarwal (Ed.), *Data Streams: Models and Algorithms* (pp. 61-84). New York, NY: Springer.

Lee, C. H., Lin, C. R., & Chen, M. S. (2005). Sliding window filtering: An efficient method for incremental mining on a time-variant database. *Information Systems*, 30(3), 227-244.

Leung, C. K. S., Khan, Q. I., & Hoque, T. (2007). CanTree: A canonical-order tree structure for incremental frequent-pattern mining. *Knowledge and Information Systems*, 11(3), 287-311.

Lin, W. Y., & Tseng, M. C. (2004). Updating generalized association rules with evolving taxonomies. In *Proceedings of 2004 International Computer Symposium*, 368-373.

Liu, J., & Yin, J. (2001). Towards efficient data remining (DRM). In D. W. L. Cheung, G. J. Williams, Q. Li (Eds.), *Proceedings of the 5th Pacific-Asia Conference on Knowledge Discovery and Data Mining***,** *Lecture Notes in Computer Science* 2035, (pp. 406-412). Springer.

Liu, G., Lu, H., & Yu, J.X. (2007). CFP-tree: A compact disk-based structure for storing and querying frequent itemsets. *Information Systems*, 32(2), 295-319.

Ma, X., Tang, S., Yang, D., & Du, X. (2002). Towards efficient re-mining of frequent patterns upon threshold changes. In X. Meng, J. Su, Y. Wang (Eds.), *Proceedings of the 3rd International Conference on Advances in Web-Age Information Management, Lecture Notes in Computer Science* 2419 (pp. 80-91). Springer.

Ng, K. K., & Lam, W. (2000). Updating of association rules dynamically. In *Proceedings of 1999 International Symposium on Database Applications in Non-Traditional Environments*, (pp. 84-91). IEEE Computer Society.

Nag, B., Deshpande, P. M., & DeWitt, D. J. (1999). Using a knowledge cache for interactive discovery of association rules. In *Proceedings of* the *5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 244-253). ACM Press.

Sarda, N. L., & Srinivas, N. V. (1998). An adaptive algorithm for incremental mining of association rules. In R. Wagner (Ed.), *Proceedings of the 9th International Workshop on Database and Expert Systems Applications* (pp. 240-245). IEEE Computer Society.

Srikant, R., & Agrawal, R. (1995). Mining generalized association rules. In U. Dayal, P. M. D. Gray, S. Nishio (Eds.), *Proceedings of the 21st International Conference on Very Large Data Bases* (pp. 407-419). Morgan Kaufmann.

Storey, V.C. (1993). Understanding semantic relationships. *Very Large Databases Journal*, 2(4), 455-488.

Thomas, S., Bodagala, S., Lsabti, K., & Ranka, S. (1997). An efficient algorithm for the incremental updation of association rules in large databases. In D. Heckerman, H. Mannila, D. Pregibon (Eds.), *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining* (pp. 263-266). AAAI Press.

Tseng, M. C., & Lin, W. Y. (2004). Maintenance of generalized association rules with multiple minimum supports. *Intelligent Data Analysis*, 8(4), 417-436.

Tseng, M. C., Lin, W. Y., & Jeng, R. (2005a). Efficient remining of generalized association Rules under multiple minimum support refinement. In R. Khosla, R. J. Howlett, L. C. Jain (Eds.), *Proceedings of the 9th International Conference on Knowledge-Based Intelligent Information & Engineering Systems, Lecture Notes in Computer Science* 3683 (pp. 1338-1344). Springer.

Tseng, M. C., Lin, W. Y., & Jeng, R. (2005b). Maintenance of generalized association rules under transaction update and taxonomy evolution. In A. M. Tjoa, J. Trujillo (Eds.), *Proceedings of the 7th International Conference on Data Warehousing and Knowledge Discovery, Lecture Notes in Computer Science* 3589 (pp. 336-345). Springer.

Tseng, M. C., Lin, W. Y., & Jeng, R. (2006). Dynamic mining of multi-supported association rules with classification ontology. *Journal of Internet Technology*, 7(4), 399-406.

KEY TERMS

Apriori-like Algorithm: A class of efficient association rule discovery algorithms that are characterized by candidate generation and pruning derived from the apriori property—an itemset can be frequent only if all of its subsets are frequent.

Association Rule: An implication rule that presents the co-occurrence of two set of items in a database. Two criteria, called support and confidence, are usually adopted to measure the quality of the rule.

Frequent Candidate Inference: A general approach aiming to determine in advance whether or not a candidate pattern is frequent according to some collected information. The purpose is to reduce unnecessary computation as well as database scanning.

Generalized Association Rule: An extension of plain association rule that presents the co-occurrence of two sets of items not only corresponding to primitive concepts in a taxonomy but also to those at higher abstraction levels.

Incremental Maintenance: A general data or knowledge maintenance strategy that refers to the process for updating previously discovered or computed knowledge without having to rebuild them as some database change occurs.

Interactive Data Mining: A human-computer collaboration knowledge discovery process that allows the user to interact with the mining system to view and discovered pattern from different perspectives by refining mining requests according to returned results.

Taxonomy: The process of naming and classifying things into groups within a larger system according to their similarities and differences.

Mining Generalized Web Data for Discovering Usage Patterns

Doru Tanasa
INRIA Sophia Antipolis, France

Florent Masseglia
INRIA Sophia Antipolis, France

Brigitte Trousse
INRIA Sophia Antipolis, France

INTRODUCTION

Web Usage Mining (WUM) includes all the Data Mining techniques used to analyze the behavior of a Web site's users (Cooley, Mobasher & Srivastava, 1999, Spiliopoulou, Faulstich & Winkler, 1999, Mobasher, Dai, Luo & Nakagawa, 2002). Based mainly on the data stored into the access log files, these methods allow the discovery of frequent behaviors. In particular, the extraction of sequential patterns (Agrawal, & Srikant, 1995) is well suited to the context of Web logs analysis, given the chronological nature of their records. On a Web portal, one could discover for example that "25% of the users navigated on the site in a particular order, by consulting first the homepage then the page with an article about the bird flu, then the Dow Jones index evolution to finally return on the homepage before consulting their personal e-mail as a subscriber". In theory, this analysis allows us to find frequent behaviors rather easily. However, reality shows that the diversity of the Web pages and behaviors makes this approach delicate. Indeed, it is often necessary to set minimum thresholds of frequency (i.e. minimum support) of about 1% or 2% before revealing these behaviors. Such low supports combined with significant characteristics of access log files (e.g. huge number of records) are generally the cause of failures or limitations for the existent techniques employed in Web usage analysis.

A solution for this problem consists in clustering the pages by topic, in the form of a taxonomy for example, in order to obtain a more general behavior. Considering again the previous example, one could have obtained: "70% of the users navigate on the Web site in a particular order, while consulting the home page then a page of news, then a page on financial indexes, then return on the homepage before consulting a service of communication offered by the Web portal". A page on the financial indexes can relate to the Dow Jones as well as the FTSE 100 or the NIKKEI (and in a similar way: the e-mail or the chat are services of communication, the bird flu belongs to the news section, etc.). Moreover, the fact of grouping these pages under the "financial indexes" term has a direct impact by increasing the support of such behaviors and thus their readability, their relevance and significance.

The drawback of using a taxonomy comes from the time and energy necessary to its definition and maintenance. In this chapter, we propose solutions to facilitate (or guide as much as possible) the automatic creation of this taxonomy allowing a WUM process to return more effective and relevant results. These solutions include a prior clustering of the pages depending on the way they are reached by the users. We will show the relevance of our approach in terms of efficiency and effectiveness when extracting the results.

BACKGROUND

The structure of a log file is formally described in Definition 7 (at the end of this chapter). This data structure can be easily transformed to the one used by sequential pattern mining algorithms. A record in a log file contains, among other data, the client IP, the date and time of the request, and the Web resource requested. To extract frequent behaviors from such a log file, for each user session in the log file, we first have to: transform the ID-Session into a client number (ID), the date and time into a time number, and the URL into an item number. Table 1 gives a file example obtained after that pre-

Table 1. File obtained after a pre-processing step

Client \ Date	d1	d2	d3	d4	d5
1	a	c	d	b	c
2	a	c	b	f	c
3	a	g	c	b	c

processing. To each client corresponds a series of times and the URL requested by the client at each time. For instance, the client 2 requested the URL "f" at time *d4*. The goal is thus, according to definition 2 and by means of a data mining step, to find the sequential patterns in the file that can be considered as frequent. The result may be, for instance, *<(a)(c)(b)(c)>* (with the file illustrated in table 1 and a minimum support given by the user: 100%). Such a result, once mapped back into URLs, strengthens the discovery of a frequent behavior, common to *n* users (with *n* the threshold given for the data mining process) and also gives the sequence of events composing that behavior.

The main interest in employing sequential patterns for Web usage mining aims at taking into account the time-dimension of the data as in the papers described thereafter.

The WUM tool (Web Utilisation Miner) proposed in (Spiliopoulou, Faulstich & Winkler, 1999) allows the discovery of navigation patterns which are interesting either from the statistical point of view or through their structure. The extraction of sequential patterns proposed by WUM is based on the frequency of the patterns considered.

Unfortunately, the amount of data (in terms of different items—pages—as well as sequences) is an issue for the techniques of sequential pattern mining. The solution would consist in lowering the minimum support used, but in this case the algorithms are not able to succeed. A proposal for solving this issue was made by the authors of (Masseglia, Tanasa & Trousse, 2004), who were interested in extracting sequential patterns with low support on the basis that high values for the minimum support often generate obvious patterns. The authors proposed to divide the problem in a recursive way in order to proceed to a phase of data mining on each sub-problem.

Another solution is to reduce the number of items by using a generalization of URLs. In (Fu, Sandhu & Shih, 2000) the authors use a syntactic generalization of URLs with a different type of analysis (clustering). Before applying a clustering, the syntactic topics of a level greater than two are replaced by their syntactic topics of a lower level.

Finally, there are other methods (Srikant & Agrawal, 1996) that can be used to extract sequential patterns by taking into account a generalization, but in other domain than the Web. Nevertheless, the automatic construction of generalizations (in the form of classes) of Web pages and for the purposes of a Web usage analysis was not studied yet. We propose in the following section such a method which is based on characteristics of Web usage data.

GWUM: MOTIVATIONS AND GENERAL PRINCIPLE

We present here our main motivations and the way we perform an usage-driven generalization of Web pages. As we mentioned in the introduction, the generalization of the items is a key factor during the extraction of sequential patterns. To understand the advantage of our work compared to a classical technique of sequential pattern mining, we propose the following example.

Let us consider the recordings from INRIA Sophia-Antipolis' Web access log (in the preprocessed form) as illustrated in Table 2. We can see there that the user C1 at date D1 made a request for the URL "homepage_DT" which is Doru Tanasa's homepage, then at date D2 he made a request for the publications page of Doru Tanasa and, finally, a request for INRIA's homepage. Similarly, user C2 made a request for Sergiu Chelcea's homepage at Date D1, and so on.

When extracting sequential patterns with a minimum support of 100%, no patterns will be found in this log (there is no item supported by 100% of the sequences). To find a frequent pattern, it will be necessary to lower the minimum support down to 50%, which allows the extraction of the following behaviors:

Table 2. Accesses to the Web site grouped by client (User)

Date / Client	D1	D2	D3
C1	homepage_DT	publications_DT	homepage_Inria
C2	homepage_SC	publications_SC	logiciels_AxIS
C3	homepage_DT	publications_AxIS	publications_DT
C4	homepage_AxIS	homepage_SC	publications_SC

1. < (homepage_DT) (publications_DT) > (supported by C1 and C3)
2. <(homepage_SC)(publications_SC)>(supported by C2 and C4)

On the other hand lowering the support:

1. Slows or even does not allow the mining algorithm to finish.
2. Discovers more general results, difficult to interpret because of their significant number and similarities (or redundancy).

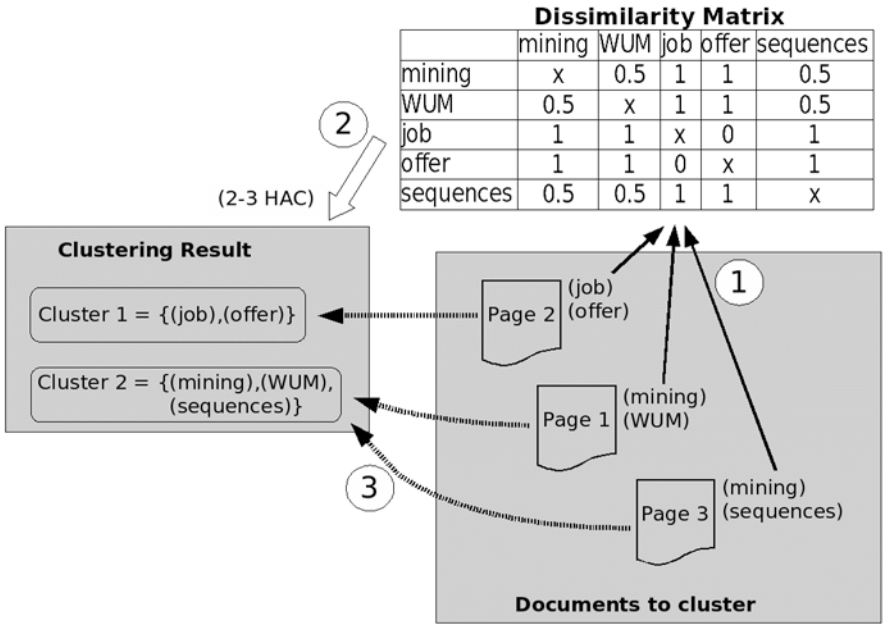
Suppose now that we are able to classify the URLs of this log in various categories. For example the category "*Pub*" would contain the pages with the researchers' publications (in our case: "publications_DT" and "publications_SC"). The category "*Mining*" would contain the pages relative to data mining (in our case, the homepages of Doru Tanasa and Sergiu Chelcea who are working on this topic). With such a classification, we would be able to extract a pattern with a support from 100% which would be: < (Mining) (Pub) >. The interpretation of this pattern is that 100% of the users consult a page about data mining and then a page of publications. We can indeed verify this behavior on the records from Table 2.

To perform a Web page generalization, we first propose to extract the information about the users' access to these pages (from the referer field). We are interested by the keywords that the user employed to search for the page. Therefore, we employ the referer of an HTTP request only when this one contains a request made to a search engine and then, we extract the keywords from the referer.

The objective is to obtain a clustering of the pages guided by their usages. Indeed, we consider that keywords which are frequently employed in a search engine to reach a page, can be used to characterize this page. For example, if the Web page of Doru Tanasa is frequently accessed with the keywords "Data Mining" and "WUM", and the Web page of Florent Masseglia with the keywords "Data Mining" and "Sequential Patterns", then it would be logical to create a category of pages "Data Mining" which would contain these two Web pages. Consequently, this means that, according to users' who reached these pages following a request on a search engine, these pages relate to "Data Mining".

The procedure that we set up for clustering the Web pages is described in Figure 1. Ideally, we should have built a matrix of dissimilarities for URLs and carry out directly a classification on it. Unfortunately, in our experiments, the high dimensionality of the data did not allow us to proceed this way. In fact, we obtained 62 721 distinct URLs, described by a total of 35 367 keywords and the matrix would have reached the size of 62 721 x 35 367 (2.2×10^9).

Figure 1. Web pages clustering based on their usage



Therefore, we chose to first classify the keywords (stage 2 in Figure 1). For that, we propose to build the matrix of distances between the keywords (stage 1 of Figure 1). The distance between two keywords *a* and *b* (i.e. *Dist(a, b)*) is given by the following formula (Jaccard's index of dissimilarity):

$$Dist(a,b) = \frac{P_{ab}}{P_a + P_b - P_{ab}}$$

where P_x is the distinct number of URLs having been reached using the keyword *x* and P_{xy} is the distinct number of URLs having been reached using the keywords *x* and *y*. The distance varies from 0 (no common page for the keywords) to 1 (same accesses for the two keywords). Once this matrix is built, we carry out the classification of the keywords, using an hierarchical clustering algorithm, the 2-3HAC (Chelcea, Bertrand & Trousse, 2003) (stage 2, in Figure 1).

Let *C* be the set of clusters for the keywords and *U* the set of URLs. The last stage consists in assigning each URL from *U* in one cluster from *C* by applying the following heuristic (stage 3 in Figure 1):

$$\forall u \in U, \forall c \in C, \forall d \in C, \ \text{if} \ \frac{words(u,c)}{|c|} \geq \frac{words(u,d)}{|d|} \ \text{then} \ c \leftarrow u$$

where *words(u, c)* is the number of keywords shared by *u* and *c*. This procedure allows assigning the pages to the clusters containing the words that describe them best. The keywords assigned to this cluster are used for its description. During our experiments, we obtained, for example, the following clusters:

- Cluster (*log,mining*), which includes for instance:
  - http://www-sop.inria.fr/axis/personnel/Florent.Masseglia/
  - http://www-sop.inria.fr/axis/personnel/Doru.Tanasa/papers/
- Cluster (*Internet,audio*), which includes for instance:
  - http://www-sop.inria.fr/rodeo/fphone/
  - http://www-sop.inria.fr/interne/accueil/audioconference.shtml

The first cluster contains pages about the following topic: data mining applied to Web access logs. The second one contains pages about the audio communications via Internet. The final step of our method now consists in replacing the URLs with their categories (when available) in the log file that will be analyzed.

EXPERIMENTS

GWUM (Generalized Web Usage Mining) is implemented in Java and is based on the PSP algorithm (Masseglia, Cathala & Poncelet, 1998). The experiments were carried out with the Web access logs of INRIA Sophia Antipolis Web site (www-sop.inria.fr) gathered in one month (October 2005). The data file has been preprocessed with the techniques described in (Tanasa & Trousse, 2004). It contains 258 061 navigations (i.e. user visits to the Web site) and 845 208 requests. The average length of navigations is 3.3 pages and there are 114 238 navigations with a length greater than 1.

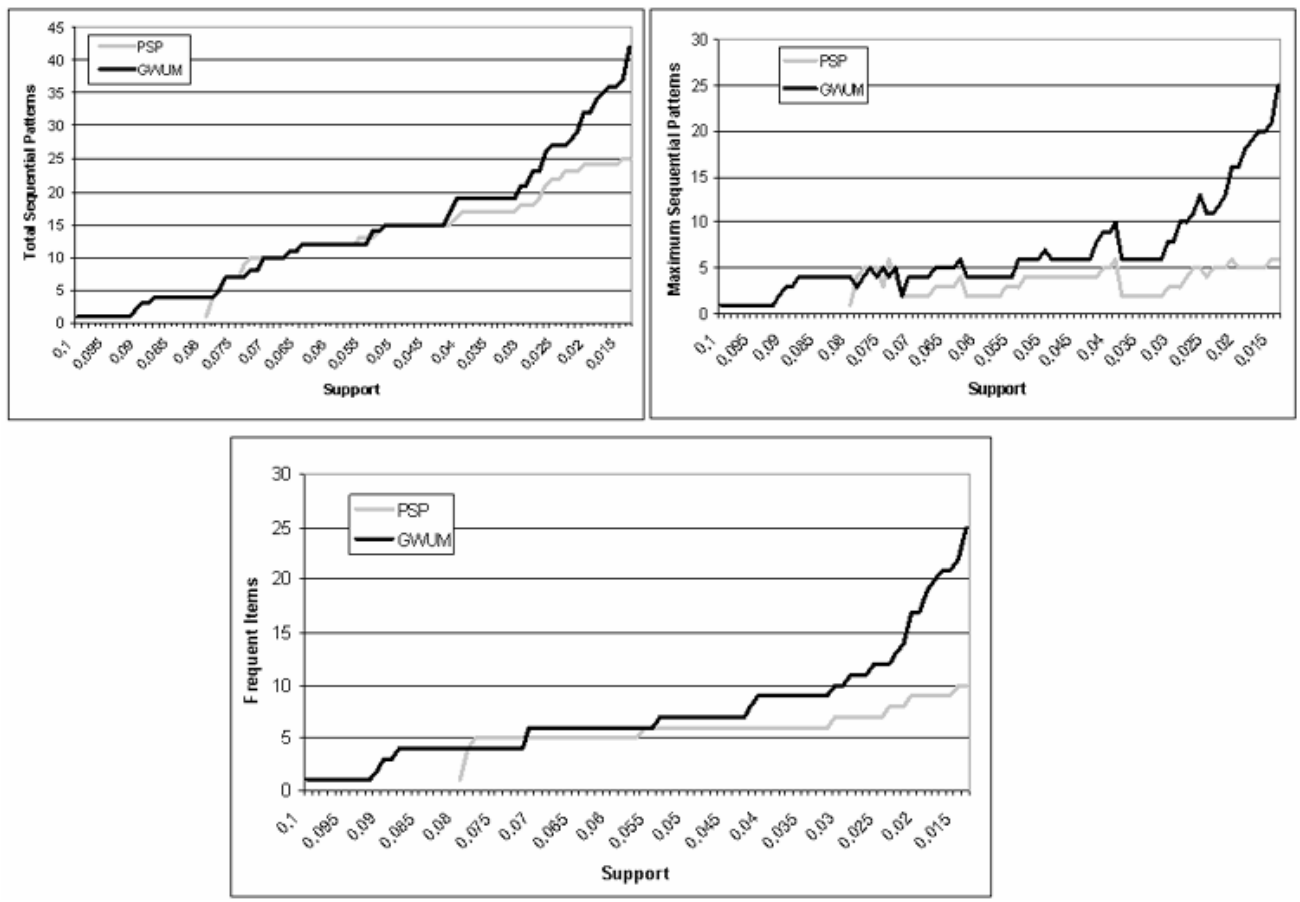
Data Preprocessing

We noticed that 164 685 requests (19.5% out of the total 845 208) were reached following a query in a search engine. From these requests we extracted 35 367 different keywords. We selected only the keywords having served for reaching at least 8 different pages. Thus, we reduced the vocabulary to 2 800 different roots of words (keywords) having been used in search engines to reach 14 264 different Web pages. We clustered the 2 800 keywords in 1 517 classes which were then utilized to classify the 14 264 URLs. For our experiments we selected only the navigations of a length higher than 1 (i.e. 114 238 navigations) together with a second dataset of generalized navigations (where the generalized URLs were replaced by their classes).

GWUM Effects on the Number of Sequential Patterns

The objective of this section is to show the benefit of our Web page generalization approach for a WUM process in terms of minimum support and number of sequential patterns extracted. Our goal is to prove that the minimum support can be increased through a generalization of URLs. Figure 2 gives the number of patterns extracted with (GWUM) and without (PSP) URLs Generalization.

Figure 2. Impact of the Generalization on the Minimum Support



The upper left part includes maximal and non-maximal patterns (without pruning). We can observe there, for example, that for the initial value of a minimum support fixed at 10%, the number of generalized patterns found was 2 whereas no patterns based on URLs only were found. In order to find the first two patterns in the URLs only dataset, it was necessary to decrease the minimum support to 7.5%. For the same minimum support, we found 4 generalized patterns using GWUM. The upper right part of Figure 2 gives the same comparison in terms of maximal patterns. From the chart, one can see that starting from a minimum support of approximately 3% the number of maximal patterns obtained is significantly higher for the generalized sequential patterns. For example, with a minimum support of 1%, the number of sequential patterns found by GWUM is 25 while PSP found only 6. Finally, in the down chart of Figure 2 we report the number of items extracted with and without URL generalization.

Example of Discovered Sequential Patterns

Throughout our experiments, we learned that most of the patterns discovered with GWUM contained classes of URLs. To illustrate the potential of our method in terms of results interpretation, but also to illustrate its impact on the minimum support, we give, in this section, one of the numerous results obtained. It arises in the following form:

< (c,code,programme,source) (software,free) >. This pattern represents the behavior of users that consulted a page about to the source code of programs written with the ***C*** language. Its support is of 0.2%. We found several possible variations for this behavior in the initial log file (without URL generalization):

1. <(www-sop.inria.fr/mimosa/fp/Bigloo/) (www-sop.inria.fr/mimosa/fp/Bigloo/doc/bigloo.

html)>. Indeed, the first page of this pattern belongs to the class (*C,code,program,source*) and the second page belongs to the class (*software,free*). Moreover, by examining these pages on the Web site, we saw that this membership was justified by the content of the pages. The support of this behavior is 0.13%.

2. <(www-sop.inria.fr/oasis/ProActive/home.html) (www-sop.inria.fr/oasis/proactive/(...)/C3DRenderingEngine.java.html})> with a support of 0.0026%.
3. And 84 patterns similar to the previous ones with a support close to zero, but whose accumulation in the classes (*C,code,program,source*) and (*software,free*) increases the support of the generalized pattern. However, none of these variations would have been found directly by using as minimum support the value of their generalized pattern's support, which emphasizes the benefit of our approach.

CONCLUSION

In this chapter, we describe an original method for Web usage analysis based on a user-driven generalization of Web pages. The information extracted for these pages, for the clustering purpose, regards the users' access to the pages. The information is obtained from the referer field of the Web access logs when the user employed a search engine to find the page. The experiment that we carried out illustrates our methodology and shows some of the benefits obtained with such an approach in the discovery of frequent sequential patterns. These benefits consist in obtaining generalized patterns with a higher support and easier to interpret. Our next objective is to explore other criteria allowing a better and more precise generalization of the Web pages. We also envisage to integrate multiple categories within the same process used for extracting sequential patterns. This will allow us to obtain patterns presenting simultaneously several categories.

ACKNOWLEDGMENT

The authors would like to thank Sergiu Chelcea and Sofiane Sellah for their support.

REFERENCES

Agrawal, R., & Srikant, R. (1995). Mining Sequential Patterns. In *Proceedings of the 11th International Conference on Data Engineering (ICDE'95)*. Philip S. Yu, Arbee L. P. Chen (Eds.) (pp. 3-14).Tapei, Taiwan. IEEE Computer Society.

Chelcea, S., Bertrand, P., & Trousse, B. (2003). A new agglomerative 2-3 hierarchical clustering algorithm. In *Innovations in Classiffication, Data Science, and Information Systems. Proceedings of the 27th Annual GfKl Conference* (pp. 3-10). University of Cottbus, Germany. Heidelberg-Berlin: Springer-Verlag

Cooley, R., Mobasher, B., & Srivastava, J. (1999). Data preparation for mining World Wide Web browsing patterns. *Knowledge and Information Systems, 1*(1), 5-32. February, 1999. Springer.

Fu, Y., Sandhu, K., & Shih, M. (2000). A generalization-based approach to clustering of web usage sessions. In *Proceedings of the 1999 KDD Workshop on Web Mining. San Diego, CA. Volume 1836 of LNAI.*. (pp.21-38). Springer-Verlag.

Masseglia, F., Cathala, F., & Poncelet, P. (1998). The PSP Approach for Mining Sequential Patterns. In *Proceedings of the 2nd European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD'98)*. LNAI (pp 176-184). Nantes, France. Springer-Verlag.

Masseglia, F., Tanasa, D., & Trousse, B. (2004). Web usage mining: Sequential pattern extraction with a very low support. *In Advanced Web Technologies and Applications: 6th Asia-Pacific Web Conference, APWeb 2004, Hangzhou, China. Proceedings. Volume 3007 of LNCS.* (pp. 513-522). Springer-Verlag

Mobasher, B., Dai, H., Luo, T., & Nakagawa, M. (2002). Discovery and evaluation of aggregate usage profiles for web personalization. *Data Mining and Knowledge Discovery 6*(1), 61-82. Springer.

Spiliopoulou, M., Faulstich, L.C., & Winkler, K. (1999). A data miner analyzing the navigational behaviour of web users. In *Proceedings of the Workshop on Machine Learning in User Modelling of the ACAI'99 International Conference*. Creta, Greece

Srikant, R., & Agrawal, R. (1996). Mining Sequential Patterns: Generalizations and Performance Improve-

ments. In *Proceedings of the 5th International Conference on Extending Database Technology (EDBT'96)*. (pp. 3-17). Avignon, France. Springer-Verlag

Tanasa, D., & Trousse, B. (2004). Advanced Data Preprocessing for Intersites Web Usage Mining. *IEEE Intelligent Systems 19*(2), 59-65. IEEE Computer Society.

DEFINITIONS

Definitions 1-3: Sequence, Data-Sequence and Subsequence.

Let $I=\{i_1, i_2, ...,i_m\}$, be a set of m literals (*items*). I is a *k-itemset* where k is the number of items in I. A **sequence** is an ordered list of itemsets denoted by $<s_1 s_2 \dots s_n>$ where s_j is an itemset. The **data-sequence** of a customer c is the sequence in D corresponding to customer c. A sequence $<a_1 a_2 \dots a_n>$ is a **subsequence** of another sequence $<b_1 b_2 \dots b_m>$ if there exist integers $i_1 < i_2 < \dots < i_n$ such that $a_1 \subseteq b_{i1}$, $a_2 \subseteq b_{i2}$, …, $a_n \subseteq b_{in}$.

Example 1. Let C be a client and $S=<(c) (d e) (h)>$, be that client's purchases. S means that "C bought item c, then he bought d and e at the same moment (i.e. in the same transaction) and finally bought item h".

Definitions 4-5: Support and Frequent Sequential Pattern.

The **support** of a sequence s, also called *supp(s)*, is defined as the fraction of total data-sequences that contain s. If $supp(s) \geq minsupp$, with a minimum support value *minsupp* given by the user, s is considered as a **frequent sequential pattern**.

Definition 6: Sequential Pattern Mining.

The problem of sequential pattern mining is to find all the frequent sequential patterns as stated in definition 2.

Definition 7: Log Entry.

A log entry describes a request made to the Web server, recorded in a chronological order into a Web server log file. For instance, in the Extended Common Log Format (ECLF), a log entry contains at least the following information:

The client's host name or its IP address,

The date and time of the request,

The operation type (GET, POST, HEAD, etc.),

The requested resource name,

The user agent (a string identifying the browser and the operating system used),

The Web access log line from Figure 3 shows that a user from the IP address 192.168.0.1, successfully requested the page /axis/people.shtml on October 3rd, 2006 at 14:05 PM. The user arrived on this page by selecting a link from the Web page http://www-sop.inria.fr/axis/ and used Microsoft Internet Explorer 6.0 to display the page. Finally the requests are sorted in ascending order of time for each couple (IP, user agent). The couple (IP, user agent) associated to the set of requests is called an user session and is given an ID-Session. For instance an user session

S12 = < (192.168.0.1, "Mozilla/4.0 …."), (['/axis/', Tue Oct 3 14:05:22] ['/axis/people.shtml', Tue Oct 3 14:05:59])> means that an user has requested URL '/axis' followed by URL '/axis/people.shtml' 37 seconds later.

Figure 3. Example of a Log Entry

```
192.168.0.1 - [03/Oct/2006:14:05:59 +0200] ``GET /axis/people.shtml HTTP/1.1'' ``Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.2; .NET CLR 1.1.4322)"
```

Mining Group Differences

Shane M. Butler
Monash University, Australia

Geoffrey I. Webb
Monash University, Australia

INTRODUCTION

Finding differences among two or more groups is an important data-mining task. For example, a retailer might want to know what the different is in customer purchasing behaviors during a sale compared to a normal trading day. With this information, the retailer may gain insight into the effects of holding a sale and may factor that into future campaigns. Another possibility would be to investigate what is different about customers who have a loyalty card compared to those who don't. This could allow the retailer to better understand loyalty cardholders, to increase loyalty revenue, or to attempt to make the loyalty program more appealing to non-cardholders.

This article gives an overview of such group mining techniques. First, we discuss two data-mining methods designed specifically for this purpose—Emerging Patterns and Contrast Sets. We will discuss how these two methods relate and how other methods, such as exploratory rule discovery, can also be applied to this task.

Exploratory data-mining techniques, such as the techniques used to find group differences, potentially can result in a large number of models being presented to the user. As a result, filter mechanisms can be a useful way to automatically remove models that are unlikely to be of interest to the user. In this article, we will examine a number of such filter mechanisms that can be used to reduce the number of models with which the user is confronted.

BACKGROUND

There have been two main approaches to the group discovery problem from two different schools of thought. The first, Emerging Patterns, evolved as a classification method, while the second, Contrast Sets, grew as an exploratory method. The algorithms of both approaches are based on the Max-Miner rule discovery system (Bayardo Jr., 1998). Therefore, we will briefly describe rule discovery.

Rule discovery is the process of finding rules that best describe a dataset. A dataset is a collection of records in which each record contains one or more discrete attribute-value pairs (or items). A rule is simply a combination of conditions that, if true, can be used to predict an outcome. A hypothetical rule about consumer purchasing behaviors, for example, might be *IF buys_milk AND buys_cookies THEN buys_cream*.

Association rule discovery (Agrawal, Imielinski & Swami, 1993; Agrawal & Srikant, 1994) is a popular rule-discovery approach. In association rule mining, rules are sought specifically in the form of where the antecedent group of items (or *itemset*), A, implies the consequent itemset, *C*. An association rule is written $A \rightarrow C$s . Of particular interest are the rules where the probability of *C* is increased when the items in *A* also occur. Often association rule-mining systems restrict the consequent itemset to hold only one item as it reduces the complexity of finding the rules.

In association rule mining, we often are searching for rules that fulfill the requirement of a minimum support criteria, *minsup*, and a minimum confidence criteria, *minconf*. Where support is defined as the frequency with which *A* and *C* co-occur support($A \rightarrow C$) = frequency($A \cup C$) and confidence is defined as the frequency with which *A* and *C* co-occur, divided by the frequency with which *A* occurs throughout all the data

$$\text{confidence}(A \rightarrow C) = \frac{\text{support}(A \rightarrow C)}{\text{frequency}(A)}$$

The association rules discovered through this process then are sorted according to some user-specified interestingness measure before they are displayed to the user.

Another type of rule discovery is *k*-most interesting rule discovery (Webb, 2000). In contrast to the support-confidence framework, there is no minimum support or confidence requirement. Instead, *k*-most interesting rule discovery focuses on the discovery of up to *k* rules that maximize some user-specified interestingness measure.

MAIN THRUST

Emerging Patterns

Emerging Pattern analysis is applied to two or more datasets, where each dataset contains data relating to a different group. An Emerging Pattern is defined as an itemset whose support increases significantly from one group to another (Dong & Li, 1999). This support increase is represented by the growth rate—the ratio of support of an itemset in group 1 over that of group 2. The support of a group G is given y:

$$\text{supp}_G(X) = \frac{\text{count}_G(X)}{|G|}$$

The *GrowthRate*(X) is defined as 0 if $\text{supp}_1(X) = 0$ and $\text{supp}_2(X) = 0$; ∞ if $\text{supp}_1(X) = 0$ and $\text{supp}_2(X) \neq 0$; or else $\text{supp}_2(X)/\text{supp}_1(X)$. The special case where *GrowthRate*(X) = ∞ is called a Jumping Emerging Pattern, as it is said to have jumped from not occurring in one group to occurring in another group. This also can be thought of as an association rule having a confidence equaling 1.0.

Emerging Patterns are not presented to the user, as models are in the exploratory discovery framework. Rather, the Emerging Pattern discovery research has focused on using the mined Emerging Patterns for classification, similar to the goals of Liu et al. (1998, 2001). Emerging Pattern mining-based classification systems include CAEP (Dong, Zhang, Wong & Li, 1999), JEP-C (Li, Dong & Ramamohanarao, 2001), BCEP (Fan & Ramamohanarao, 2003), and DeEP (Li, Dong, Ramamohanarao & Wong, 2004). Since the Emerging Patterns are classification based, the focus is on classification accuracy. This means no filtering method is used, other than the infinite growth rate constraint used during discovery by some the classifiers (e.g., JEP-C and DeEP). This constraint discards any Emerging Pattern X for which *GrowthRate*(X) $\neq \infty$.

Contrast Sets

Contrast Sets (Bay & Pazzani, 1999, 2001) are similar to Emerging Patterns, in that they are also itemsets whose support differs significantly across datasets. However, the focus of Contrast Set research has been to develop an exploratory method for finding differences between one group and another that the user can utilize, rather than as a classification system focusing on prediction accuracy. To this end, they present filtering and pruning methods to ensure only the most interesting and optimal number rules are shown to the user, from what is potentially a large space of possible rules.

Contrast Sets are discovered using STUCCO, an algorithm that is based on the Max-Miner search algorithm (Bayardo Jr., 1998). Initially, only Contrast Sets are sought that have supports that are both significant and the difference large (i.e., the difference is greater than a user-defined parameter, *mindev*). Significant Contrast Sets (*cset*), therefore, are defined as those that meet the criteria:

$$P(cset \mid G_i) \neq P(cset \mid G_j)$$

Large Contrast Sets are those for which:

$$\text{support}(cset, G_i) - \text{support}(cset, G_j) \geq mindev$$

As Bay and Pazzani have noted, the user is likely to be overwhelmed by the number of results. Therefore, a filter method is applied to reduce the number of Contrast Sets presented to the user and to control the risk of type-1 error (i.e., the risk of reporting a Contrast Set when no difference exists). The filter method employed involves a chi-square test of statistical significance between the itemset on one group to that Contrast Set on the other group(s). A correction for multiple comparisons is applied that lowers the value of α as the size of the Contrast Set (number of attribute value pairs) increases.

Further pruning mechanisms also are used to filter Contrast Sets that are purely specializations of other more general Contrast Sets. This is done using another chi-square test of significance to test the difference between the parent Contrast Set and its specialization Contrast Set.

Mining Group Differences Using Rule Discovery

Webb, Butler, and Newlands (2003) studied how Contrast Sets relate to generic rule discovery approaches. They used the OPUS_AR algorithm-based Magnum Opus software to discover rules and to compare them to those discovered by the STUCCO algorithm.

OPUS_AR (Webb, 2000) is a rule-discovery algorithm based on the OPUS (Webb, 1995) efficient search technique, to which the Max-Miner algorithm is closely related. By limiting the consequent to a group variable, this rule discovery framework is able to be adapted for group discovery.

While STUCCO and Magnum Opus specify different support conditions in the discovery phase, their conditions were proven to be equivalent (Webb et al., 2003). Further investigation found that the key difference between the two techniques was the filtering technique. Magnum Opus uses a binomial sign test to filter spurious rules, while STUCCO uses a chi-square test. STUCCO attempts to control the risk of type-1 error by applying a correction for multiple comparisons. However, such a correction, when given a large number of tests, will reᵃ uce the value to an extremely low number, meaning that the risk of type-2 error (i.e., the risk of not accepting a non-spurious rule) is substantially increased. Magnum Opus does not apply such corrections so as not to increase the risk of type-2 error.

While a chi-square approach is likely to be better suited to Contrast Set discovery, the correction for multiple comparisons, combined with STUCCO's minimum difference, is a much stricter filter than that employed by Magnum Opus. As a result of Magnum Opus' much more lenient filter mechanisms, many more rules are being presented to the end user. After finding that the main difference between the systems was their control of type-1 and type-2 errors via differing statistical test methods, Webb, et al. (2003) concluded that Contrast Set mining is, in fact, a special case of the rule discovery task.

Experience has shown that filters are important for removing spurious rules, but it is not obvious which of the filtering methods used by systems like Magnum Opus and STUCCO is better suited to the group discovery task. Given the apparent tradeoff between type-1 and type-2 error in these data-mining systems, recent developments (Webb, 2003) have focused on a new filter method to avoid introducing type-1 and type-2 errors. This approach divides the dataset into exploratory and holdout sets. Like the training and test set method of statistically evaluating a model within the classification framework, one set is used for learning (the exploratory set) and the other is used for evaluating the models (the holdout set). A statistical test then is used for the filtering of spurious rules, and it is statistically sound, since the statistical tests are applied using a different set. A key difference between the traditional training and test set methodology of the classification framework and the new holdout technique is that many models are being evaluated in the exploratory framework rather than only one model in the classification framework. We envisage the holdout technique will be one area of future research, as it is adapted by exploratory data-mining techniques as a statistically sound filter method.

Case Study

In order to evaluate STUCCO and the more lenient Magnum Opus filter mechanisms, Webb, Butler, and Newlands (2003) conducted a study with a retailer to find interesting patterns between transactions from two different days. This data was traditional market-basket transactional data, containing the purchasing behaviors of customers across the many departments. Magnum Opus was used with the group, encoded as a variable and the consequent restricted to that variable only.

In this experiment, Magnum Opus discovered all of the Contrast Sets that STUCCO found, and more. This is indicative of the more lenient filtering method of Magnum Opus. It was also interesting that, while all of the Contrast Sets discovered by STUCCO were only of size 1, Magnum Opus discovered conjunctions of sizes up to three department codes.

This information was presented to the retail marketing manager in the form of a survey. For each rule, the manager was asked if the rule was surprising and if it was potentially useful to the organization. For ease of understanding, the information was transformed into a plain text statement.

The domain expert judged a greater percentage of the Magnum Opus rules of surprise than the STUCCO contrasts; however, the result was not statistically significant. The percentage of rules found that potentially were useful were similar for both systems. In this case, Magnum Opus probably found some rules that were

spurious, and STUCCO probably failed to discover some rules that were potentially interesting.

FUTURE TRENDS

Mining differences among groups will continue to grow as an important research area. One area likely to be of future interest is improving filter mechanisms. Experience has shown that the use of filter is important, as it reduces the number of rules, thus avoiding overwhelming the user. There is a need to develop alternative filters as well as to determine which filters are best suited to different types of problems.

An interestingness measure is a user-generated specification of what makes a rule potentially interesting. Interestingness measures are another important issue, because they attempt to reflect the user's interest in a model during the discovery phase. Therefore, the development of new interestingness measures and determination of their appropriateness for different tasks are both expected to be areas of future study.

Finally, while the methods discussed in this article focus on discrete attribute-value data, it is likely that there will be future research on how group mining can utilize quantitative, structural, and sequence data. For example, group mining of sequence data could be used to investigate what is different about the sequence of events between fraudulent and non-fraudulent credit card transactions.

CONCLUSION

We have presented an overview of techniques for mining differences among groups, discussing Emerging Pattern discovery, Contrast Set discovery, and association rule discovery approaches. Emerging Patterns are useful in a classification system where prediction accuracy is the focus but are not designed for presenting the group differences to the user and thus don't have any filters.

Exploratory data mining can result in a large number of rules. Contrast Set discovery is an exploratory technique that includes mechanisms to filter spurious rules, thus reducing the number of rules presented to the user. By forcing the consequent to be the group variable during rule discovery, generic rule discovery software like Magnum Opus can be used to discover group differences. The number of differences reported to the user by STUCCO and Magnum Opus are related to the different filter mechanisms for controlling the output of potentially spurious rules. Magnum Opus uses a more lenient filter than STUCCO and thus presents more rules to the user. A new method, the holdout technique, will be an improvement over other filter methods, since the technique is statistically sound.

REFERENCES

Agrawal, R., Imielinski, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, Washington, D.C., USA.

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules. *Proceedings of the 20th International Conference on Very Large Data Bases*, Santiago, Chile.

Bay, S.D., & Pazzani, M.J. (1999). Detecting change in categorical data: Mining contrast sets. *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Diego, USA.

Bay, S.D., & Pazzani, M.J. (2001). Detecting group differences: Mining contrast sets. *Data Mining and Knowledge Discovery, 5*(3), 213-246.

Bayardo, Jr., R.J. (1998). Efficiently mining long patterns from databases. *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*, 85-93, Seattle, Washington, USA.

Dong, G., & Li, J. (1999). Efficient mining of emerging patterns: Discovering trends and differences. *Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining*, San Diego, California, USA.

Dong, G., Zhang, X., Wong, L., & Li, J. (1999). CAEP: Classification by aggregating emerging patterns. *Proceedings of the Second International Conference on Discovery Science*, Tokyo, Japan.

Fan, H., & Ramamohanarao, K. (2003). A Bayesian approach to use emerging patterns for classification. *Proceedings of the 14th Australasian Database Conference*, Adelaide, Australia.

Li, J., Dong, G., & Ramamohanarao, K. (2001). Making use of the most expressive jumping emerging patterns for classification. *Knowledge and Information Systems, 3*(2), 131-145.

Li, J., Dong, G., Ramamohanarao, K., & Wong, L. (2004). DeEPs: A new instance-based lazy discovery and classification system. *Machine Learning, 54*(2), 99-124.

Liu, B., Hsu, W., & Ma, Y. (1998). Integrating classification and association rule mining. *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, New York, New York.

Liu, B., Ma, Y., & Wong, C.K. (2001). Classification using association rules: Weaknesses and enhancements. In V. Kumar et al. (Eds.), *Data mining for scientific and engineering applications* (pp. 506-605). Boston: Kluwer Academic Publishing.

Webb, G.I. (1995). An efficient admissible algorithm for unordered search. *Journal of Artificial Intelligence Research, 3*, 431-465.

Webb, G.I. (2000). Efficient search for association rules. *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Boston, Massachesetts, USA.

Webb, G.I. (2003). Preliminary investigations into statistically valid exploratory rule discovery. *Proceedings of the Australasian Data Mining Workshop*, Canberra, Australia.

Webb, G.I., Butler, S.M., & Newlands, D. (2003). On detecting differences between groups. *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Washington, D.C., USA.

KEY TERMS

Association Rule: A rule relating two itemsets—the antecedent and the consequent. The rule indicates that the presence of the antecedent implies that the consequent is more probable in the data. Written as $A \rightarrow C$.

Contrast Set: Similar to an Emerging Pattern, it is also an itemset whose support differs across groups. The main difference is the method's application as an exploratory technique rather than as a classification one.

Emerging Pattern: An itemset that occurs significantly more frequently in one group than another. Utilized as a classification method by several algorithms.

Filter Technique: Any technique for reducing the number of models with the aim of avoiding overwhelming the user.

Growth Rate: The ratio of the proportion of data covered by the Emerging Pattern in one group over the proportion of the data it covers in another group.

Holdout Technique: A filter technique that splits the data into exploratory and holdout sets. Rules discovered from the exploratory set then can be evaluated against the holdout set using statistical tests.

Itemset: A conjunction of items (attribute-value pairs (e.g., *age = teen* $\wedge$ *hair = brown*).

k-Most Interesting Rule Discovery: The process of finding *k* rules that optimize some interestingness measure. Minimum support and/or confidence constraints are not used.

Market Basket: An itemset; this term is sometimes used in the retail data-mining context, where the itemsets are collections of products that are purchased in a single transaction.

Rule Discovery: The process of finding rules that then can be used to predict some outcome (e.g., *IF 13 <= age <= 19 THEN teenager*).

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 795-799, copyright 2005 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

Mining Repetitive Patterns in Multimedia Data

Junsong Yuan
Northwestern University, USA

Ying Wu
Northwestern University, USA

INTRODUCTION

One of the focused themes in data mining research is to discover frequent and repetitive patterns from the data. The success of frequent pattern mining (Han, Cheng, Xin, & Yan, 2007) in structured data (e.g., transaction data) and semi-structured data (e.g., text) has recently aroused our curiosity in applying them to multimedia data. Given a collection of unlabeled images, videos or audios, the objective of repetitive pattern discovery is to find (if there is any) similar patterns that appear repetitively in the whole dataset. Discovering such repetitive patterns in multimedia data brings in interesting new problems in data mining research. It also provides opportunities in solving traditional tasks in multimedia research, including visual similarity matching (Boiman & Irani, 2006), visual object retrieval (Sivic & Zisserman, 2004; Philbin, Chum, Isard, Sivic & Zisserman, 2007), categorization (Grauman & Darrell, 2006), recognition (Quack, Ferrari, Leibe & Gool, 2007; Amores, Sebe, & Radeva, 2007), as well as audio object search and indexing (Herley, 2006).

- In image mining, frequent or repetitive patterns can be similar image texture regions, a specific visual object, or a category of objects. These repetitive patterns appear in a sub-collection of the images (Hong & Huang, 2004; Tan & Ngo, 2005; Yuan & Wu, 2007, Yuan, Wu & Yang, 2007; Yuan, Li, Fu, Wu & Huang, 2007).
- In video mining, repetitive patterns can be repetitive short video clips (*e.g.* commercials) or temporal visual events that happen frequently in the given videos (Wang, Liu & Yang, 2005; Xie, Kennedy, Chang, Divakaran, Sun, & Lin, 2004; Yang, Xue, & Tian, 2005; Yuan, Wang, Meng, Wu & Li, 2007).
- In audio mining, repetitive patterns can be repeated structures appearing in music (Lartillot, 2005) or broadcast audio (Herley, 2006).

Repetitive pattern discovery is a challenging problem because we do not have any *a prior* knowledge of the possible repetitive patterns. For example, it is generally unknown in advance (i) what the repetitive patterns look like (e.g. shape and appearance of the repetitive object/contents of the repetitive clip); (ii) where (location) and how large (scale of the repetitive object or length of the repetitive clip) they are; (iii) how many repetitive patterns in total and how many instances each repetitive pattern has; or even (iv) whether such repetitive patterns exist at all. An exhaustive solution needs to search through all possible pattern sizes and locations, thus is extremely computationally demanding, if not impossible.

BACKGROUND

As a purely unsupervised task, repetitive pattern discovery is different from traditional pattern detection/retrieval problem, where a pattern template or a query example is provided and the task is to find its re-occurrences (Divakara, Peker, Chang, Radhakrishnan, & Xie, 2004). When mining repetitive patterns, we have no *a prior* knowledge of the pattern. Therefore, it differs from supervised learning problems (*e.g.* classification and retrieval) that have been widely studied in machine learning research.

Repetitive pattern discovery is also different from unsupervised tasks like clustering. The task of clustering is to partition a collection of data samples into several disjoint groups, where data samples belonging to the same group are similar to each other, whereas data samples from different groups are dissimilar. Instead of clustering individual data samples, pattern discovery aims at repetitive patterns that are shared by some data samples. Such a repetitive pattern describes the common characteristics among the data. For example, given an image dataset, a repetitive pattern can be a sub-image region (e.g. a visual object) that appears in

many images. Thus the repetitive pattern corresponds to an image region while not the whole image.

MAIN FOCUS

Multimedia data mining is different from transaction and text data mining mainly in two aspects. Firstly, transaction and text data are discrete data, for example, transactions are composed of items and texts are composed of vocabularies. However, multimedia data are usually characterized by continuous features and these features generally exhibit much larger variabilities and uncertainties than predefined items and vocabularies. Taking the visual pattern in image data for example, the same visual pattern (e.g. a visual object) is likely to exhibit quite different visual appearances under different lighting conditions, views, scales, not to mention partial occlusion. As a result, it is very difficult find invariant visual features that are insensitive to these variations to uniquely characterize visual patterns.

Another characteristic of multimedia data mining, besides the features uncertainties, is that multimedia patterns have more complex structure than transaction and text patterns. For example, the spatial configuration is important in characterizing an image pattern. Therefore, when discovering repetitive patterns from images, the difficulty of representing and discovering spatial patterns prevents straightforward generalization of traditional frequent pattern mining methods that are applicable to transaction data, which are orderless sets of items, or text data, which are represented by strings of words. Although there exist methods for spatial collocation pattern discovery from geo-spatial data (Huang, Shekhar, & Xiong, 2004), they cannot be directly applied to image data which are characterized by high-dimensional continuous features. Similarly, when mining audio and video data, we should take the temporal structure into consideration, as the repetitive patterns are usually temporal sequences.

The main focus of repetitive pattern discovery in multimedia data mining is to develop methods that address the aforementioned two challenges. In the following, we summarize three major components of repetitive pattern discovery in multimedia data: (1) pattern representation, (2) pattern matching and (3) pattern mining. We will discuss the specific challenges and the-state-of-the-art techniques in each component.

Pattern Representation

Pattern representation is important for general machine learning and data mining tasks. In multimedia data mining, it determines how we represent the image/video/audio data, typically in terms of features and data models. Data mining algorithms perform directly on the extracted features instead of the raw multimedia data. Due to the page limit, we do not list all possible features that can be extracted from the multimedia data. Instead, we discuss the requirements of the features for repetitive pattern discovery.

- For repetitive pattern discovery in image data, it is usually desirable that the extracted visual features are robust under different pattern variations like rotation, scale changes and even affine transformations. Recently, it is found that the "bag of features" representation is beneficial for pattern discovery because it can handle pattern variations as well as incomplete patterns. The basic visual features are local visual primitives detected from images (Philbin, Chum, Isard, Sivic & Zisserman, 2007; Quack, Ferrari, Leibe & Gool, 2007; Sivic & Zisserman, 2004; Yuan, Wu, & Yang, 2007). Each visual primitive describes an informative local image region and is expected to be invariant under the following variations. (Mikolajczyk, Tuytelaars, Schmid, Zisserman, Matas, Schaffalitzky, Kadir, & Gool, 2005).

1. **View point changes:** The same image pattern can look different under different view-points, depending on the capture angle of the camera.
2. **Scale changes:** The same image pattern can look smaller or larger, depending on the distance of the camera to the object, as well as the zoom in/out parameters of the camera.
3. **Lighting condition changes:** The appearances of a same image pattern may vary according to different lighting conditions at the capture.
4. **Partial Occlusions:** It is possible that a pattern (e.g. a repetitive object) is occluded in the image. Such an incomplete pattern, however, could still be of interests and want to be discovered.

5. **Intra-class variations:** When the goal of pattern discovery is to find objects of the same category, it is important that the features inherit the ability to handle intra-class variations. That is, the objects of the same category share similar features, while the objects of different categories do not.

- For repetitive pattern discovery in audio and video data, features are extracted based on video/audio frames or temporal windows. Usually features are desirable to be robust under various signal changes.

1. **Speed changes:** It is possible that the same temporal pattern may be played at different speeds due to frame rate changes or pattern variations.
2. **Codec changes:** It is possible the same contents are compressed into different format/quality after coding.
3. **Intra-class variations:** When the goal of pattern discovery is to find similar patterns of the same category, it is important that the feature inherit the ability to handle intra-class variations.

Pattern Matching

Pattern matching plays a critical role in multimedia data mining, as its accuracy and speed largely determine the quality of the data mining results as well as the efficiency of the data mining methods. Given two patterns, pattern matching determines whether they are similar or not. Therefore it can be applied to measure the repetition of a query pattern. In matching multimedia patterns, it is important to consider the spatial/spatial-temporal characteristics of the image/video data (Yuan & Wu, 2007). We discuss pattern matching methods for image and video/audio respectively.

- In image pattern discovery, to avoid handling high-dimensional features, many recent work tend to represent an image as a "bag of words" by quantizing high-dimensional visual features into "visual items" or "visual words" through clustering (*e.g.* k-means clustering). Therefore each image is translated to a visual document or a set of transaction records (Philbin, Chum, Isard, Sivic, & Zisserman, 2007; Quack, Ferrari, Leibe, & Gool, 2007; Sivic J. & Zisserman, 2004; Yuan, Wu, & Yang, 2007), with each transaction corresponding to a local image patch and describing its composition of visual primitive classes (items). Matching images is like matching documents/transactions, and data mining techniques can be applied to such an induced transaction database from images for discovering meaningful visual patterns.
- For the video/audio case, video frames (e.g. key frames) or video/audio temporal windows can be quantized into a discrete symbol and the whole video sequence becomes a symbolic string which is similar to a DNA sequence. Traditional sequence mining algorithm can thus be applied (Wang, Liu, & Yang, 2005). Although the quantization/clustering idea is popular in the literature, some recent work also argue that such representations discard information due to the invertible quantization errors from continuous visual features to discrete visual words. Therefore matching visual primitives in the original continuous domain is still preferred as it gives more accurate matching results (Philbin, Chum, Isard, Sivic & Zisserman, 2007; Yuan & Wu, 2007). Many methods performed directly on original continuous features, unfortunately, are *not* computationally efficient due to the cost of matching two sets of points in the high-dimensional space (Boiman & Irani, 2006; Hong & Huang, 2004; Tan & Ngo, 2005). To overcome the matching cost of image data, approximate matching methods of linear complexity are proposed (Grauman & Darrell, 2006; Yuan & Wu, 2007).

To speed up the pattern matching process, many efficient indexing and searching methods are applied, such as the inverted file index method (Sivic & Zisserman, 2004), locality sensitive hashing for approximate nearest neighbor search (Yuan, Li, Fu, Wu, & Huang, 2007), as well as fast similarity matching in the high-dimensional space (Grauman & Darrell, 2006; Yuan & Wu, 2007).

Pattern Mining

Although a repetitive pattern is easy to be evaluated, for example, we can determine whether a query pattern is repetitive by simply searching it through the whole dataset and counting the re-occurrences, it is

difficult to discover repetitive patterns automatically. Generally, pattern mining is of high computational cost, not only due to the large size of the database, but also because we have no *a prior* knowledge of the possible repetitive patterns. One advantage of the "bag of words" representation is that we can easily transfer the multimedia data into transaction data, thus efficient data mining algorithm like frequent itemset mining can be applied directly (Yuan, Wu & Yang, 2007). However, if continuous features are applied, due to the huge candidate number of repetitive patterns, efficient pruning strategies are required to avoid exhaustive search of repetitive patterns (Yuan, Li, Fu, Wu & Huang, 2007).

FUTURE TRENDS

By mining repetitive patterns in multimedia data, it will bring advantages in many tasks associated with multimedia database, such as content-based retrieval, indexing and visual object detection and categorization. We are looking forward to more applications of repetitive pattern discovery from multimedia data. At the same time, to handle real large database (*e.g.* image database of billions images), more efficient repetitive pattern discovery algorithm is required. Finally, interactive repetitive pattern discovery can be an interesting direction which takes consideration of the feedback from the users in order to improve the pattern mining results.

CONCLUSION

As an emerging research direction, repetitive pattern discovery provides opportunities in solving many existing problems in multimedia research. Compared with traditional data mining in transaction data, we need to handle the feature uncertainty and pattern complexity problems, which are challenges brought by the multimedia data.

There are three major components in mining repetitive patterns from multimedia data: pattern representation, pattern matching and pattern mining. As a cross-discipline research topic, it needs more efforts from multimedia database, data mining, machine learning, information retrieval, and computer vision. As a powerful data mining tool, the ultimate objective of repetitive pattern discovery in multimedia data is to help analyze, organize and retrieve the multimedia database more efficiently and accurately.

REFERENCES

Amores J., Sebe N., & Radeva P. (2007). Context-Based Object-Class Recognition and Retrieval by Generalized Correlograms, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 29(10): 1818-1833.

Boiman O. & Irani M. (2006). Similarity by composition, in *Proc. Neural Information Processing Systems*.

Divakaran A., Peker K.-A., Chang S.-F., Radhakrishnan R., & Xie L. (2004). Video Mining: Pattern Discovery versus Pattern Recognition, in *Proc. IEEE Conf. on Image Processing*.

Grauman K. & Darrell T. (2006). Unsupervised learning of categories from sets of partially matching image features, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*.

Han J., Cheng H., Xin D., & Yan X. (2007). Frequent pattern mining: current status and future directions, *Data Mining and Knowledge Discovery*, 1(14).

Herley C. (2006). ARGOS: Automatically extracting repeating objects from multimedia streams. *IEEE Trans. on Multimedia*, 8(1): 115–129.

Hong P. & Huang T.-S. (2004). Spatial pattern discovery by learning a probabilistic parametric model from multiple attributed relational graphs, *Discrete Applied Mathematics*, 1(3). 113–135.

Huang Y., Shekhar S., & Xiong H. (2004). Discovering collocation patterns from spatial data sets: a general approach. *IEEE Transaction on Knowledge and Data Engineering*, 16(12).1472–1485.

Lartillot O. (2005). Efficient Extraction of Closed Motivic Patterns in Multi-Dimensional Symbolic Representations of Music, in *IEEE/WIC/ACM Intl. Conf. on Web Intelligence*, 229-235

Mikolajczyk K., Tuytelaars T., Schmid C., Zisserman A., Matas J., Schaffalitzky F., Kadir T., & Gool L. V. (2005). A comparison of affine region detectors. *Intl. Journal of Computer Vision*, 65(1/2), 43-72.

Philbin J., Chum O., Isard M., Sivic J., & Zisserman A. (2007). Object retrieval with large vocabularies and fast spatial matching, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*.

Sivic J. & Zisserman A. (2004). Video data mining using configurations of viewpoint invariant regions, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*.

Tan K.-K. & Ngo C.-W. (2005). Common pattern discovery using earth mover's distance and local flow maximization, in *Proc. IEEE Intl. Conf. on Computer Vision*.

Quack T., Ferrari V., Leibe B. & Gool L. (2007). Efficient mining of frequent and distinctive feature configurations, in *Proc. Intl. Conf. on Computer Vision*.

Wang P., Liu Z.-Q., & Yang S.-Q. (2005). A probabilistic template-based approach to discovering repetitive patterns in broadcast videos. in *Proc. ACM Intl. Conf. on Multimedia*.

Xie L., Kennedy L., Chang S.-F., Divakaran A., Sun H., & Lin C.-Y. (2004). Discovering meaningful multimedia patterns with audio-visual concepts and associated text. In *Proc. IEEE Conf. on Image Processing*.

Yang X., Xue P., & Tian Q. (2005) A repeated video clip identification system. *Proc. ACM Intl. Conf. on Multimedia*.

Yuan J. & Wu Y. (2007). Spatial random partition for common visual pattern discovery, in *Proc. Intl. Conf. on Computer Vision.*

Yuan J., Li Z., Fu Y., Wu Y., Huang T.-S. (2007). Common spatial pattern discovery by efficient candidate pruning, in *Proc. IEEE Conf. on Image Processing.*

Yuan J., Wu Y., & Yang M. (2007). From frequent itemsets to semantically meaningful visual patterns, in *Proc. ACM Intl. Conf. on Knowledge Discovery and Data Mining.*

Yuan J., Wang W., Meng J., Wu Y. & Li D. (2007). Mining repetitive clips through finding continuous paths, in *Proc. ACM Intl. Conf. on Multimedia.*

KEY TERMS

Frequent Itemset Mining (FIM): Data mining process for discovering itemsets that appear frequent enough from a large collection of transactions.

High-Dimensional Feature Space: An abstract vector space of high-dimension, typically above tens of dimensions. In pattern recognition, each pattern sample is represented as a point in the feature space. The number of dimensions is equal to the number of features.

Invertible Quantization Error: The inaccuracy introduced in the process of quantization, which depends on the resolution of the quantization and cannot be recovered after quantization.

Locality Sensitive Hashing (LSH): A fast approximate nearest neighbor search technique in the high-dimensional space through random projection.

Multimedia Data Mining: The process of discovering novel knowledge from multimedia data, such as images, audios and videos, which can help to analyze, organize and retrieve the multimedia database.

Nearest Neighbor Search: The process of finding nearest neighbors of a given query point in the feature space. Nearest neighbor search is useful in retrieval and classification.

Quantization: The process that approximates a continuous value/vector with a discrete value/symbol.

Scale of a Visual Pattern: The relative size/length of the pattern appearing in images/videos.

Transaction Data: A specific type of data, where each transaction is composed by a collection of discrete items. A typical example of transaction data is the market basket data, where each single transaction is composed by the collection of items purchased by the customer. Transaction data is popularly considered in data mining.

Visual Primitive: The elementary visual pattern. For images, it denotes informative local interesting points or regions. For audios and videos, it can be the basic unit, like a shot or a fixed-length window.

Mining Smart Card Data from an Urban Transit Network

Bruno Agard
École Polytechnique de Montréal, Canada

Catherine Morency
École Polytechnique de Montréal, Canada

Martin Trépanier
École Polytechnique de Montréal, Canada

INTRODUCTION

In large urban areas, smooth running public transit networks are key to viable development. Currently, economic and environmental issues are fueling the need for these networks to adequately serve travel demand, thereby increasing their competitiveness and their market share. Better balance between transit supply and demand will also help reduce and control operating costs.

The fact is, however, that transit operators are finding it extremely difficult to adjust the service to meet the demand, because this demand changes continuously with the time or day of travel (period of the day, day of the week, season or holiday) and other factors like weather and service breakdown. In order to enhance their service, operators need to better understand the travel demand (customer behaviors and the variability of the demand in space and time). This can be achieved only by continuously monitoring the day-to-day activities of users throughout the transit network.

Some large cities around the world take advantage of smart card capabilities to manage their transit networks by using Smart Card Automated Fare Collection Systems (SCAFCS). An SCAFCS gives travelers greater flexibility, since a single card may be used by one user at various times and on different parts of the transit network, and may support various fare possibilities (by travel, line, zone, period, etc.). For transit operators, these systems not only validate and collect fares, but also represent a rich source of continuous data regarding the use of their network. Actually, this continuous dataset (developed for fare collection) has the potential to provide new knowledge about transit use. Following the application of various pretreatments which make it possible to extract real-time activity, data mining techniques can reveal interesting patterns. These techniques are aimed at precisely describing customer behavior, identifying sets of customers with similar behaviors, and measuring the spatial and temporal variability of transit use. Patterns are extracted and analyzed to document various issues, such as identifying transit use cycles or homogeneous days and weeks of travel for various periods of the year. This information is required for a better understanding and modeling of customer behavior, and consequently better adjustment of the service to the demand. These adjustments may, for instance, lead to the restructuring of the transit network, to the adaptation of route scheduling or to the definition of new and different subscription options (fares).

Below, results from various experiments conducted with a real dataset are provided. They show the potential of data mining to provide useful and novel information about user behavior on a transit network. The data processed in the study are extracted from a system operating in a Canadian city (Gatineau, Quebec).

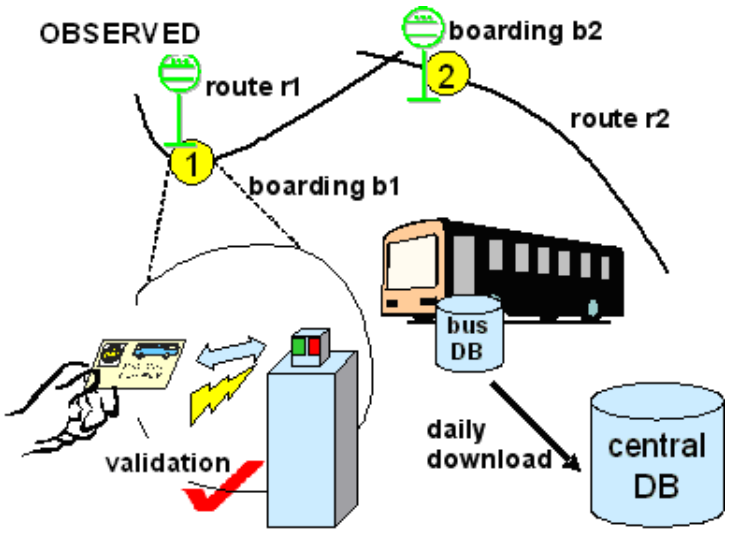
BACKGROUND

Smart Card in Public Transport

Generally, SCAFCS are composed of cards, onboard readers and a centralized information system (see Figure 1).

A smart card is simply an RFID device implanted in a transport card, and is similar to a typical credit card. Smart cards offer various advantages over traditional paper systems:

Figure 1. Smart card automated fare collection system (SCAFCS)



- Validation of the user transaction (boarding/transfer) is instantaneous and does not require any interaction with the driver.
- Complex fare systems (multiple zones, rates), common in large metropolitan areas where networks are integrated, can easily be managed.
- Elementary security procedures can be taken.
- Data are continuously collected, resulting in datasets much larger than those usually available to measure customer traffic; hence, a larger user proportion is observed.

These systems are being used increasingly frequently, since the software and hardware tools necessary to support their implementation are now stable and accessible, with a good quality of data output (Chira-Chavala and Coifman 1996, Meadowcroft 2005). Smart cards in transportation have mainly been implemented to simplify fare collection, but could also be used to advantage to monitor the service itself. In most cases, the adoption of a smart card system by transit authorities is related to their level of funding and the level of sophistication of their other technologies (Iseki et al. 2007, Yoh et al. 2006). Cheung (2006) demonstrated the overall benefits of smart card systems in The Netherlands, but it was Bagchi and White (2004, 2005) who were the first to substantiate this potential for transit planning. Using three case studies (British networks), they illustrate the ability of smart card data to estimate turnover rates, trip rates per card and the impacts of the use of smart cards on the number of linked trips. They also discuss the complementary nature of smart card data and other data collection methods, arguing that smart cards should not replace those methods. Utsunomiya et al. (2006) presented a study based on Chicago Transit Authority data for a one-week period involving about 500,000 boarding transactions. They reported difficulties associated with the data, especially missing transactions and incorrect bus routes.

Travel Behavior Variability

In practice, many transit planners use statistics from synthetic models, onboard travel surveys or regional travel surveys to describe how their networks are used. Average statistics are constructed, describing typical customer behaviors during a typical weekday. The underlying hypothesis that all weekdays are similar is

less well accepted. Actually, many studies validate the fact that travel behaviors vary a great deal in space and time. Some studies show the importance of understanding the variations in daily peak profiles to arriving at a better assessment of demand management schemes (Bonsall et al. 1984). Others illustrate the construction of detailed results by classifying travelers in terms of similar daily activity patterns (Jun and Goulias 1997). Garling and Axhausen (2003) talk about the habitual nature of travel. Metrics are available to evaluate any similarity between days of travel using a six-week travel diary (Schlich and Axhausen 2003), and others for the spatio-temporal variability (time-space prism) of day-to-day behaviors (Kitamura et al. 2006).

Not only is there agreement that variability needs to be assessed, but also that it is hard to measure, because the available data usually rely on a single day's record of each individual's travel. Bagchi and White (2004) have argued that a SCAFCS could improve these results because they provide access to larger sets of individual data. They also offer the possibility of linking user and transit card information. Hence, continuous data are available for long periods of time, which may lead to better knowledge of a large number of transit users.

More recent results have proved the ability of smart card data to measure the demand precisely and to understand its dynamics with a view to making day-to-day predictions (Morency et al. 2007). The methods developed will soon provide automatic survey tools which will help planners perform these tasks. However, trip-end analyses require the estimation of the destination of smart card trips. Trepanier et al. (2007) propose a destination estimation model with a success rate of about 80% at peak hours.

It is important to keep in mind that a transportation smart card belongs to an individual user, and that privacy rules must be respected (Clarke 2001, CNIL 2003).

MAIN FOCUS

The case study data used in our investigation were provided by the Société de transport de l'Outaouais (STO), a transit authority which manages a fleet of 200 buses in the Gatineau (Quebec) region of Canada (240,000 inhabitants). At this time (2007), about 80% of all STO passengers have a smart card, and every STO bus is equipped with a smart card reader and a GPS capturing device.

Data Collection Process

Figure 2 shows the general data flow of the smart card information system. Each time a smart cardholder boards a bus (step A), a transaction is recorded. The reader validates the smart card with the help of basic operational information, like the route number, the direction and the fare policy for this route. The current location is punched in at the same time (GPS reading). Every night, when the bus returns to the depot, all this data on transactions is uploaded to the database server, called the *Système d'Information et de Validation des Titres* (SIVT). At this point, the operational data for the next day are downloaded to the reader on the bus (step B).

The SIVT server collects data on both user and boarding (transactions), but keeps the information separate for privacy purposes. The SIVT server receives its operational data (routes, run assignments, stop list) from the STO's service operation information system (step C).

The SIVT exchanges data with the STO's accounting system on a continuous basis (step D). The accounting system is responsible for issuing and "recharging" individual smart cards.

In an SCAFCS, data are designed for fare collection and revenue management. Basic statistics, such as the number of users per line or per day, are easily extracted. The hypothesis driving the work reported here is that the data hold a great deal of other information about the overall system, and that it may provide interesting and relevant knowledge. That knowledge may be useful for helping planners better understand transit user behavior, leading to improved service.

The volume of data is perpetually growing, and this represents a considerable challenge in terms of knowledge extraction. In the present case, over a 10-month period, January 1st to October 4th, 2005, 27,033 smart cards were being used on the network, and nearly 6.2 millions transactions were validated. Trepanier and Chapleau (2001) developed an object model based on the Transportation Object-Oriented Modeling approach (Figure 3). The model shows strong links between network elements (routes, stops), operational data (drivers, vehicles, work pieces) and user data (cards, transactions, fares). There are up to 36 fare types at the STO, characterized by user status (student, adult, elderly) and network use privileges (regular, express and interzone routes).

Figure 2. General data flow of the smart card information system

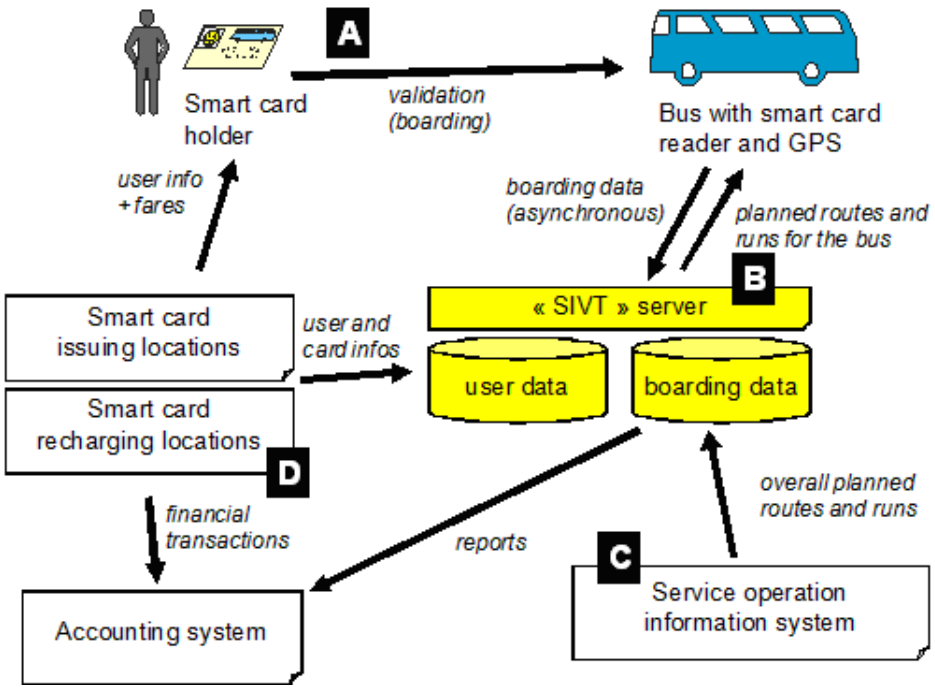
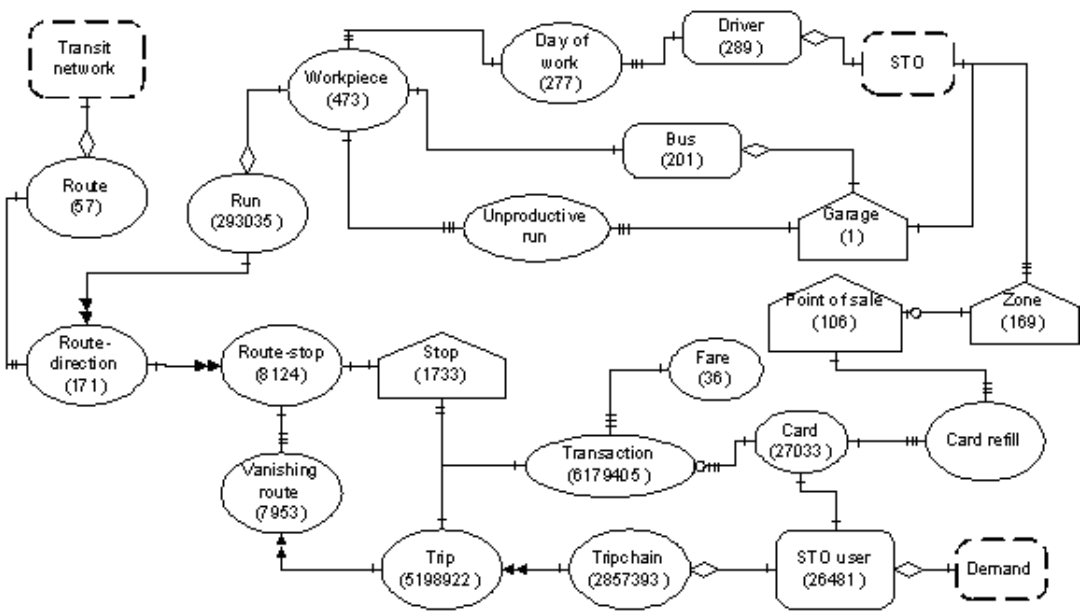


Figure 3. STO smart card object model for January to October 2005 data



Mining the Data

Here, we focus on the automatic extraction of user patterns on the transit network, using data mining tools. Data transformation and analysis are explained in detail. For a description of any of the data mining tools we mention, the reader may refer to the relevant chapters of the present encyclopedia.

Data Preprocessing

Each data refers to a specific spatial location and time. Every time a card is validated, a new line is created in the database. Since one card corresponds to one user, multiple lines can be created for each individual user. Each user may also have a different number of boardings, depending on his or her own activity.

Preprocessing operations have been conducted in such a way that every customer is represented on a fixed-length vector, in order to be able to extract user behavior in terms of temporal behavior. Depending on the accuracy of the analysis, two transformations are presented, both based on the same idea. In one, a day is made up of 24 periods (one hour each), and in the other a week is made up of 28 periods (4 periods per day, 7 days). Also included is a card ID and the day or week concerned (see Figure 4).

Group Behavior

Our analysis focuses on the way the individual users behave on the transit network, in terms of temporal activity. Different "groups" of users are identified. The number of groups depends on the level of granularity of interest in the study. An example, a split in 4 clusters with a k-mean, is presented in Figure 5.

Analysis of each cluster provides interesting information about the way the users who compose it use the transit network (temporal behavior). The following results are observed (see Figure 6, Agard et al. 2006): Cluster 1 represents 45.6% of the user weeks; Cluster 2: 14.8%; Cluster 3: 14.3%; and Cluster 4: 25.2%. Almost 60% of the cards from type "Adult" are in Cluster 1, while almost 80% of "Elderly" cards

Figure 4. Data transformation for the extraction of temporal user behavior

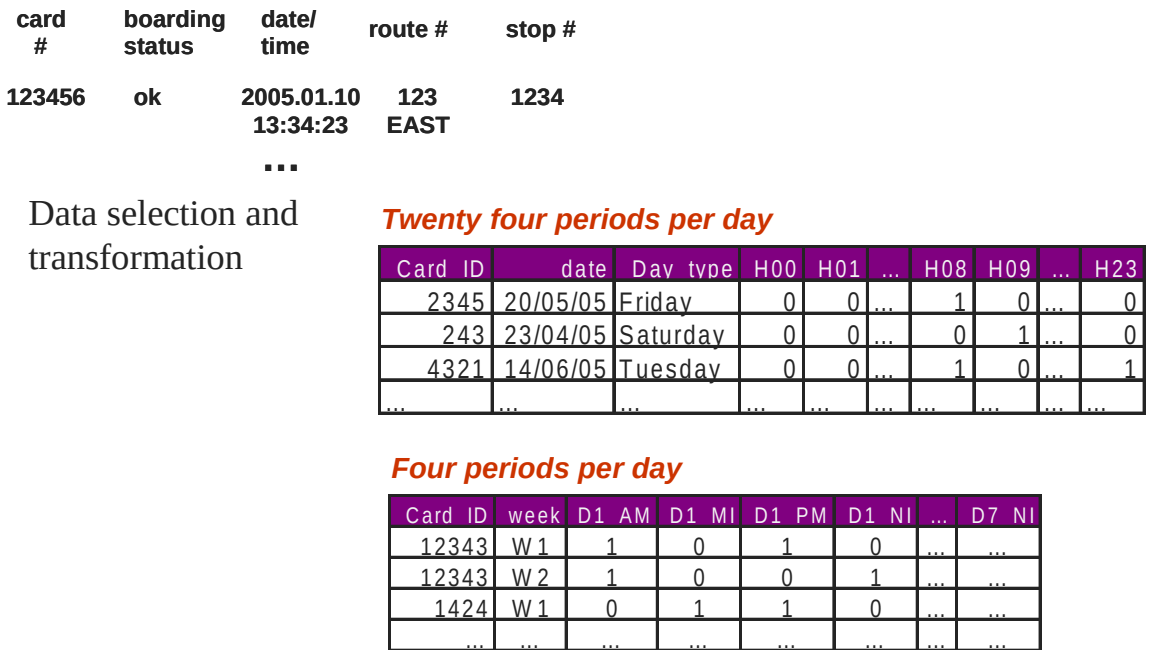


Figure 5. Decomposition into 4 clusters with a k-mean

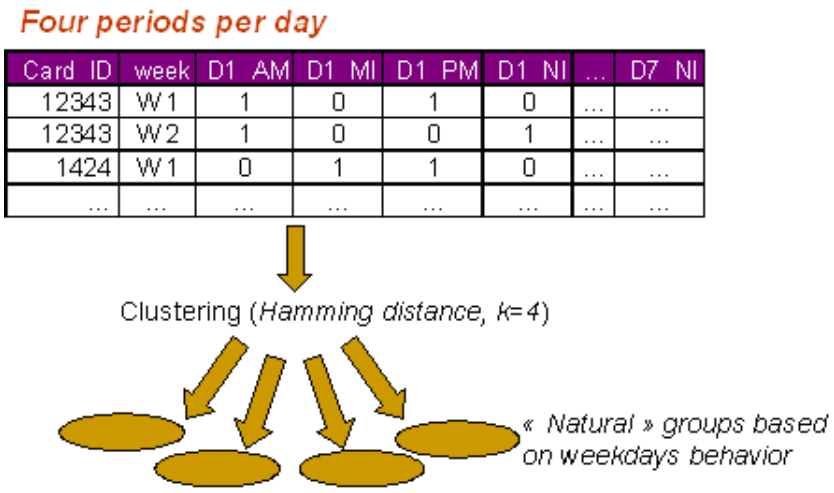


Figure 6. Composition of the clusters

Card type	Cl 1	Cl 2	Cl 3	Cl 4	TOT
Adult	58.8%	13.9%	9.2%	18.1%	100%
Student	21.0%	17.7%	26.4%	34.8%	100%
Elderly	6.2%	6.4%	7.9%	79.5%	100%

Card type	Cl 1	Cl 2	Cl 3	Cl 4
Adult	85.6%	62.4%	42.7%	47.7%
Student	13.9%	36.1%	55.4%	41.7%
Elderly	0.5%	1.4%	1.8%	10.6%
Total	100%	100%	100%	100%

are in Cluster 4. In contrast, Cluster 1 is composed of 85.6% of "Adults", Cluster 3 of 55.4% of "Students" and Cluster 4 of 10% of "Elderly".

Some results for the temporal behavior for each cluster (Agard et al. 2006) are summarized in Figure 7.

The users in Cluster 1 are typical workers, 79.4% of them traveling during the peak AM hours and 71.0% during the peak PM hours on weekdays. The users in Cluster 3 are the earlybirds, 77.6% of them traveling during the peak AM hours and 74.8% during the mid-day period. Clusters 2 and 4 show no clear patterns, but similar behavior may be observed from one weekday to another. However, Cluster 4 users are characterized by light use of the transit network.

Figure 7. Temporal behavior for clusters 1 to 4

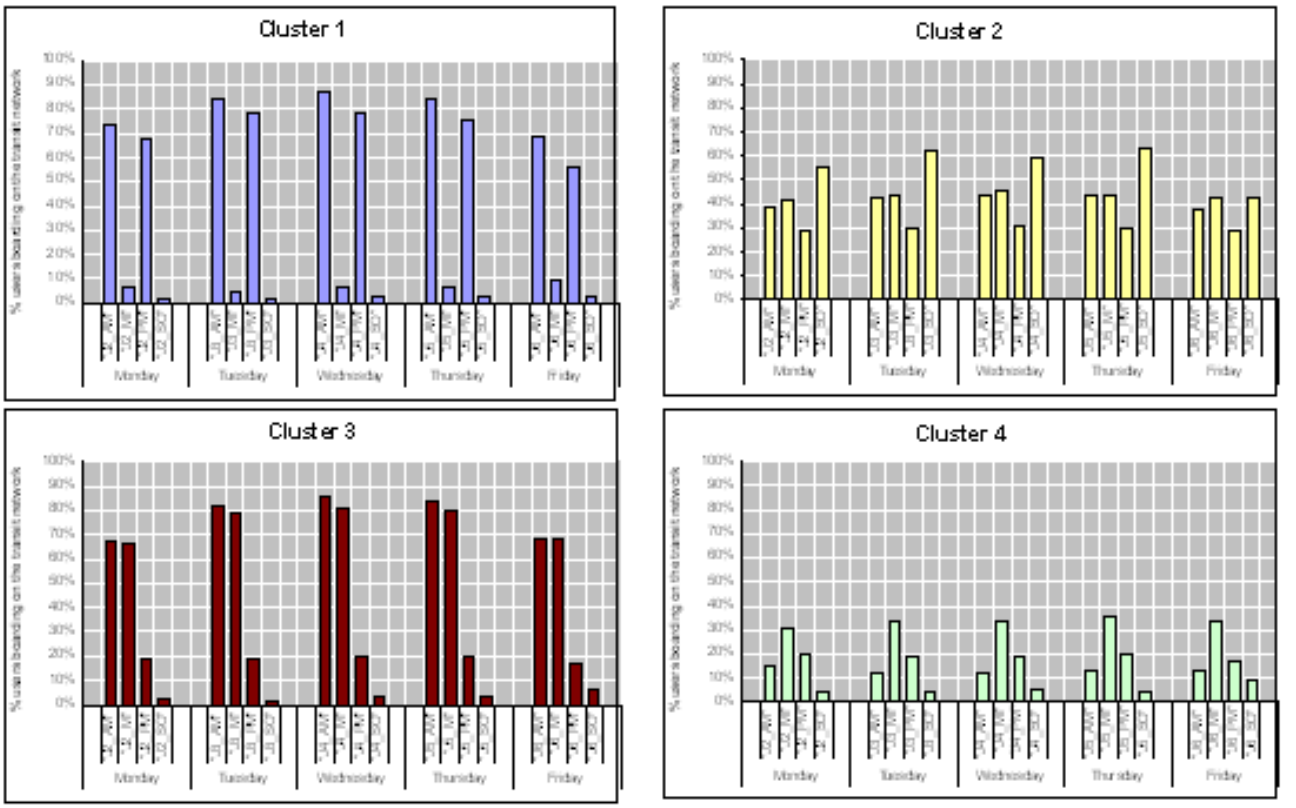
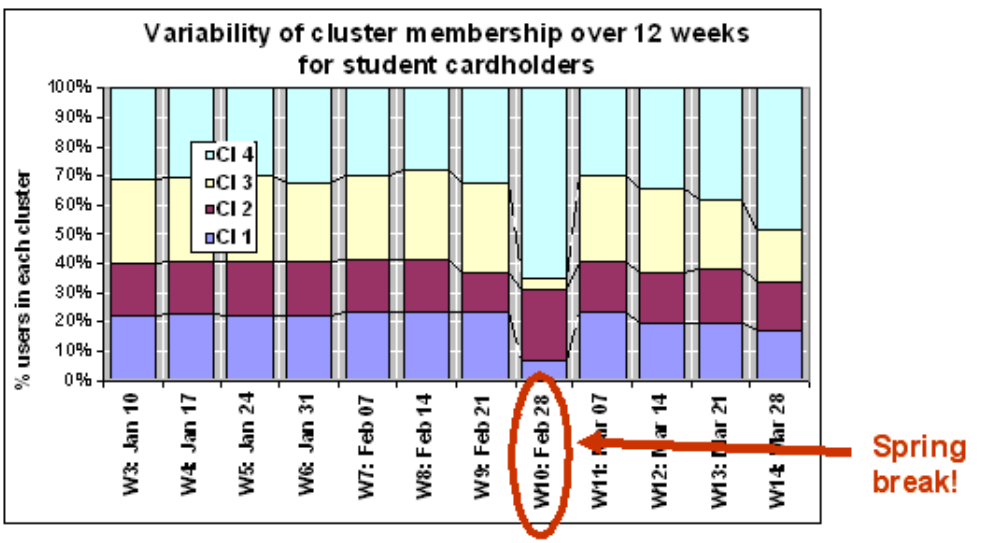


Figure 8. Variability of student behavior



If we focus on the variability (Agard et al. 2006) over 12 weeks (January to April, 2005) of a predefined population (students), we observe that the average proportion of each behavior is relatively stable, except during the spring break where Cluster 4 (light use) predominates (also, Cluster 2, with no clear pattern, becomes important during this period). All this shows a major change in student behaviors (Figure 8).

Individual Behavior

Application of the same technique (clustering on temporal description of user behavior) at various levels of granularity may reveal individual patterns (Figure 9).

Clusters are computed on the overall population in order to extract patterns which may be useful for all the datasets. From these clusters, it is relatively easy to construct one map for each user to represent the variability between clusters for each type of day (Morency et al. 2006) (see Figure 10).

The behavior of the card owner ("Regular ADULT" above) is easy to predict. For example, since 95% of this user's Sunday behavior belongs to the same cluster (no. 9), we can easily predict, from the description of cluster no. 9, which contains an hourly record of boarding times, this user's transit behavior on Sundays. Other users behave less predictably, and, in these instances, each day may be characterized by less representative clusters.

FUTURE TRENDS

Data mining of a SCAFC is a powerful tool for providing information about transit system use. Further studies will make it possible to evaluate travel behaviors in space, and to detect the existence of punctual activities involving new transit paths or established paths which are evolving.

Current and future research concerns include the following:

- Refinement of the alighting location estimation model (to be implemented at the STO)
- Geospatial trip behavior (using geospatial data mining techniques)
- Specific route usage over space and time (to measure user turnover)
- More detailed mining in a specific time period (using the exact time of the first boarding for each day over a one-year period, for example)
- Application of mining techniques to the measurement of the spatial variability of travel behavior (boarding locations) and linking this information with spatial statistics

Figure 9. k-mean clustering on daily activities

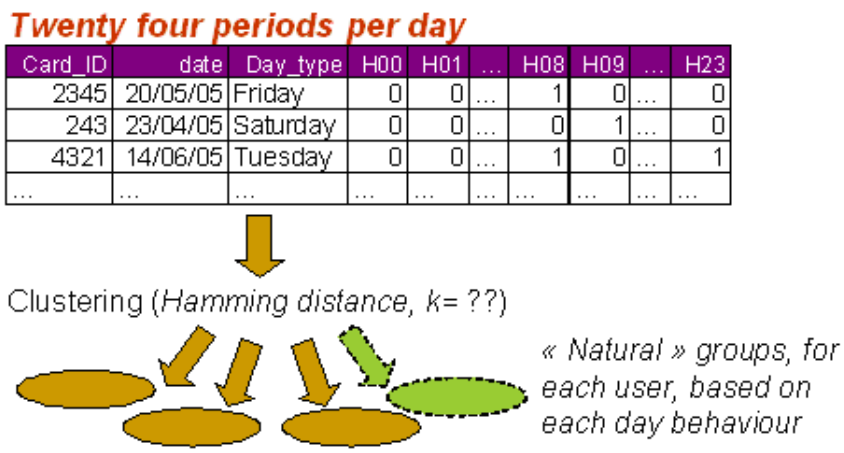
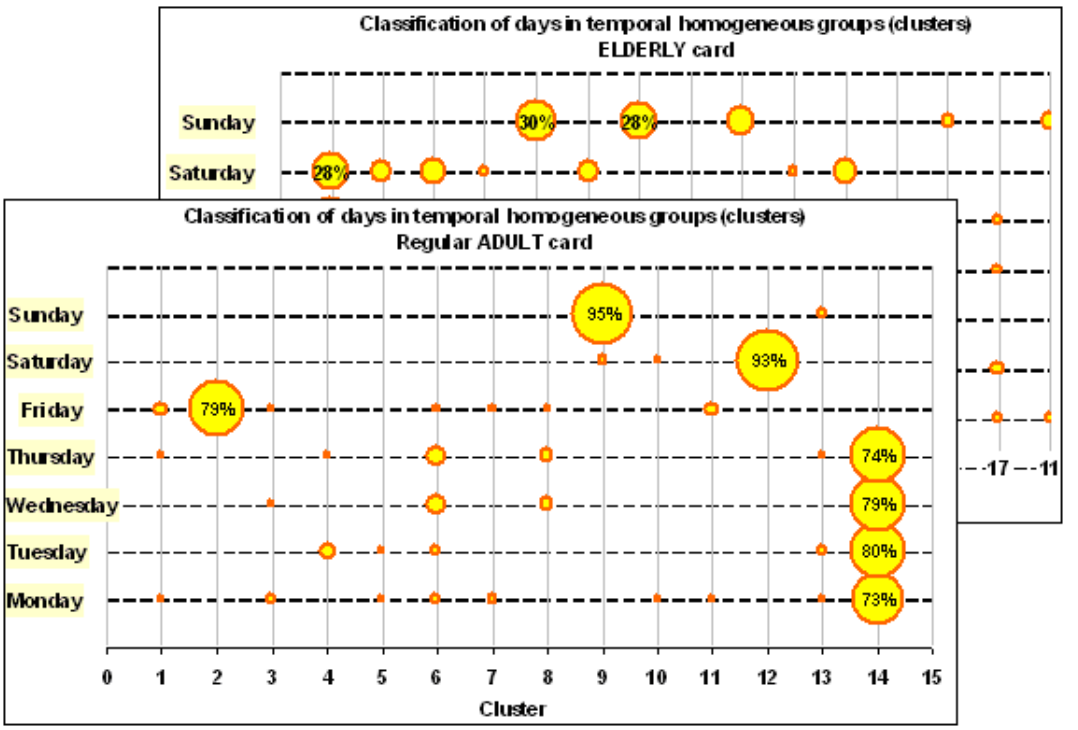


Figure 10. Examples of maps of behavior variability



- Application of subgroup discovery to reveal specific patterns in user behaviors.

CONCLUSION

This chapter shows that data mining is a powerful tool for knowledge extraction from a SCAFCS. Used for more than fare collection, an SCAFCS has the potential to reveal hidden patterns which are useful for addressing operational concerns.

The analysis of continuous data makes it possible to extract user transit behaviors and their variability. Data mining tools allow us to obtain interesting information about the overall system. That knowledge will help planners better understand the behaviors of transit users, with a view to improving service.

Trip behavior is an important topic in the transportation literature, and understanding user behavior helps in the design of better networks offering better services.

In the study presented in this chapter, the data generated for fare collection were used in different ways to evaluate the actual use of a public transit system, making it possible to:

- define typical customers and measure their travel behaviors, and
- analyze the variability of use of the system with respect to day, week or season.

With these results, it is possible to improve a transit system by adjusting the balance between customer behaviors and the level of service provided on each route. It is also possible to propose variable fares, which encourage temporal or spatial shifts towards less congested runs and routes.

Smart card data are proving to be richer than the usual travel data that focus on a single day of travel for a single individual. They also constitute a supplementary source for analysis.

REFERENCES

Agard, B., Morency, C., Trépanier, M. (2006) Mining public transport user behaviour from smart card data, *12th IFAC Symposium on Information Control Problems in Manufacturing – INCOM 2006*, Saint-Etienne, France, May 17–19.

Bagchi, M., White, P.R. (2004) What role for smart-card data from a bus system? *Municipal Engineer 157*, March, 39-46.

Bagchi, M., White, P.R. (2005) The potential of public transport smart card data, *Transport Policy*, 12, 464-474.

Bonsall, P., Montgomery, F., Jones, C. (1984) Deriving the Constancy of Traffic Flow Composition from Vehicle Registration Data, *Traffic Engineering and Control*, 25(7/8), 386-391.

Cheung, F. (2006) Implementation of Nationwide Public Transport Smart Card in the Netherlands, *Transportation Research Record*, no. 1971, 127-132.

Chira-Chavala, T., Coifman, B. (1996) Effects of Smart Cards on Transit Operators, *Transportation Research Record*, no. 1521, 84-90.

Clarke, R. (2001). Person location and person tracking: Technologies, risks and policy implications, *Information Technology & People*, 14(2), 206-231.

CNIL –Commission nationale de l'informatique et des libertés (2003) Recommandation relative à la collecte et au traitement d'informations nominatives par les sociétés de transports collectifs dans le cadre d'applications billettiques, *CNIL*, Délibération n° 03-038.

Gärling, T., Axhausen, K.W. (2003) Introduction: Habitual travel choice, *Transportation*, 30(1), 1-11.

Iseki, H., Yoh, A.C., Taylor, B.D. (2007) Are Smart Cards the Smart Way to Go? Examining the Adoption of Smart Card Fare Systems Among U.S. Transit Agencies, *Transportation Research Board Meeting*, Washington, 22p.

Jun, M, Goulias, K. (1997) A dynamic analysis of person and household activity and travel patterns using data from the first two waves in the Puget Sound Transportation Panel, *Transportation*, no. 24, 309-331.

Kitamura, R., Yamamoto, T., Susilo, Y.O., Axhausen, K.W. (2006) How routine is a routine? An analysis of the day-to-day variability in prism vertex location, *Transportation Research Part A*, no. 40, 259-279.

Meadowcroft, P. (2005) Hong Kong raises the bar in smart card innovation, *Card Technology Today*, 17(1), 12-13.

Morency C., Trépanier M., Agard B. (2006) Analysing the variability of transit users behaviour with smart card data, *The 9th International IEEE Conference on Intelligent Transportation Systems – ITSC 2006*, Toronto, Canada, September 17-20.

Morency C., Trépanier M., Agard B. (2007) Measuring transit use variability with smart card data, *Transport Policy*, 14(3), 193-203.

Schlich, R, Axhausen, K.W. (2003) Habitual travel behaviour: Evidence from a six-week travel diary, *Transportation*, no. 30, 13-36.

Trépanier, M., Chapleau, R. (2001) Analyse orientée-objet et totalement désagrégée des données d'enquêtes ménages origine-destination, *Revue canadienne de génie civil*, Ottawa, 28(1), 48-58.

Trépanier, M., Chapleau, R., Tranchant, N. (2007) Individual Trip Destination Estimation in Transit Smart Card Automated Fare Collection System, *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations*, 11(1), 1-15.

Utsunomiya, M., Attanucci, J., Wilson, N. (2006) Potential Uses of Transit Smart Card Registration and Transaction Data to Improve Transit Planning, *Transportation Research Record*, no. 1971, 119–126.

Yoh, A.C., Iseki, H., Taylor, B.D., King, D.A. (2006) Interoperable Transit Smart Card Systems: Are We Moving Too Slowly or Too Quickly? *Transportation Research Record*, no. 1986, 69–77.

KEY TERMS

Activity Pattern: Sequence of activities made by a single person within a day. Each activity is bounded by the trips made before and after.

Board: To go onto or into a transportation vehicle.

Smart Card: Electronic device the size of a credit card which can store a small amount of data. It acts essentially like a radiofrequency identification tag (RFID).

Smart Card Automated Fare Collection System: System used to collect and validate fare payment aboard public transit vehicles.

Transit Mode: Public mean of transportation like bus, subway, commuter train.

Travel Behavior: In the field of transportation research, refers to the trip habits (frequency, purpose, time of departure, trip-end locations, etc.) of individual users on each transportation mode.

Urban Transit Network: Mass transit system which serves an urban population. It comprises modes and the provision of the service is guaranteed in the form of fixed schedules.

Mining Software Specifications

David Lo
National University of Singapore, Singapore

Siau-Cheng Khoo
National University of Singapore, Singapore

INTRODUCTION

Software is a ubiquitous component in our daily life. It ranges from large software systems like operating systems to small embedded systems like vending machines, both of which we frequently interact with. Reducing software related costs and ensuring correctness and dependability of software are certainly worthwhile goals to pursue.

Due to the short-time-to-market requirement imposed on many software projects, documented software specifications are often lacking, incomplete and outdated (Deelstra, Sinnema & Bosch 2004). Lack of documented software specifications contributes to difficulties in understanding existing systems. The latter is termed program comprehension and is estimated to contribute up to 45% of total software cost which goes to billions of dollars (Erlikh 2000, Standish 1984; Canfora & Cimitile 2002; BEA 2007). Lack of specifications also hampers automated effort of program verification and testing (Ammons, Bodik & Larus 2002).

One solution to address the above problems is mining (or automatic extraction of) software specification from program execution traces. Given a set of program traces, candidate partial specifications pertaining to the behavior a piece of software obeys can be mined.

In this chapter, we will describe recent studies on mining software specifications. Software specification mining has been one of the new directions in data mining (Lo, Khoo & Liu 2007a, Lo & Khoo 2007). Existing specification mining techniques can be categorized based on the form of specifications they mine. We will categorize and describe specification mining algorithms for mining five different target formalisms: Boolean expressions, automata (Hopcroft, Motwani & Ullman 2001), Linear Temporal Logic (Huth & Ryan 2003), frequent patterns (Han & Kamber 2006) and Live Sequence Charts (Harel & Marelly 2003).

BACKGROUND

Different from many other engineering products, software changes often during its lifespan (Lehman & Belady 1985). The process of making changes to a piece of software e.g., to fix bugs, to add features, etc., is known as software maintenance. During maintenance, there is a need to understand the current version of the software to be changed. This process is termed as program comprehension. Program comprehension is estimated to take up to 50% of software maintenance efforts which in turn is estimated to contribute up to 90% of total software costs (Erlikh 2000, Standish 1984; Canfora & Cimitile 2002). Considering the $216.0 billion of software component contribution to the US GDP at second quarter 2007, the cost associated with program comprehension potentially goes up to billions of dollars (BEA 2007). One of the root causes of this problem is the fact that documented software specification is often missing, incomplete or outdated (Deelstra, Sinnema & Bosch 2004). Mining software specifications is a promising solution to reduce software costs by reducing program comprehension efforts.

On another angle, software dependability is a well sought after goal. Ensuring software runs correctly at all times and identifying bugs are two major activities pertaining to dependability. Dependability is certainly an important issue as incorrect software has caused the loss of billions of dollars and even the loss of lives (NIST 2002; ESA & CNES 1996; GAO 1992). There are existing tools for performing program verification. These tools take formal specifications and automatically check them against programs to discover inconsistencies, identify bugs or ensure that all possible paths in the program satisfy the specification (Clarke, Grumberg & Peled 1999). However, programmers' reluctance and difficulty in writing formal specifications have been some of the barriers to the widespread adoption

of such tools in the industry (Ammons, Bodik & Larus 2002, Holtzmann 2002). Mining software specifications can help to improve software dependability by providing these formal specifications automatically to these tools.

MAIN FOCUS

There are a number of specification mining algorithms available. These algorithms can be categorized into families based on the target specification formalisms they mine. These include specification miners that mine Boolean expressions (Ernst, Cockrell, Griswold and Notkin 2001), automata (Cook & Wolf 1998; Reiss & Reinieris, 2001; Ammons, Bodik & Larus 2002; Lo & Khoo 2006a; Lo & Khoo 2006b; Mariani, Papagiannakis and Pezzè 2007; Archaya, Xie, Pei & Xu, 2007; etc.), Linear Temporal Logic expressions (Yang, et al. 2006; Lo, Khoo & Liu 2007b; Lo, Khoo & Liu 2008, etc.), frequent patterns (Li & Zhou, 2005; El-Ramly, Stroulia & Sorenson, 2002; Lo, Khoo & Liu 2007a; etc.) and Live Sequence Charts (Lo, Maoz & Khoo 2007a, Lo, Maoz & Khoo 2007b).

These mined specifications can aid programmers in understanding existing software systems. Also, a mined specification can be converted to run-time tests (Mariani, Papagiannakis & Pezzè 2007; Lo, Maoz & Khoo 2007a; Lo, Maoz & Khoo 2007b) or input as properties-to-verify to standard program verification tools (Yang, Evans, Bhardwaj, Bhat and Das, 2006; Lo, Khoo & Liu 2007b).

Preliminaries

Before proceeding further, let us describe some preliminaries. Specifications can be mined from either traces or code. A program trace is a sequence of events. Each event in a trace can correspond to a statement being executed, or a method being called, etc. In many work, an event is simply the signature of a method that is being called. Traces can be collected in various ways. A common method is to instrument a code by inserting `print' statement to various locations in the code. Running the instrumented code will produce a trace file which can then be analyzed.

Mining Boolean Expressions

Ernst, Cockrell, Griswold and Notkin (2001) propose an algorithm that mines Boolean expressions from program execution traces at specific program points. Sample Boolean expressions mined are x=y+z, x>5, etc. The algorithm is based on a set of templates which is then matched against the program execution traces. Template instances that are satisfied by the traces above a certain threshold are outputted to the user.

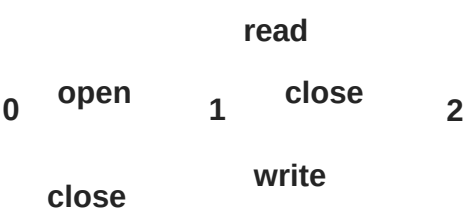
Mining Automata

Simply put, an automaton is a labeled transition system with start and end nodes. Traversing an automaton from start to end nodes will produce a sentence, which will correspond to a program behavior (e.g., file protocol: open-read-write-close). An automaton represents a set of valid sentences that a program can behave. An example of an automaton representing a file protocol is drawn in Figure 1.

One of the pioneering work on mining automata is the work by Ammons, Bodik and Larus (2002). In their work, a set of pre-processed traces are input to an automata learner (Raman & Patrick, 1997). The output of the learner is a specification in the form of an automaton learned from the trace file. This automaton is then presented to end users for fine tuning and modifications.

Lo and Khoo (2006a) define several metrics for assessing the quality of specification mining algorithms that mine automata. Among these metrics, precision and recall are introduced as measures of accuracy to existing specification miners producing automata. Precision refers to the proportion of sentences accepted by the language described by the mined automaton that are also accepted by the true specification. Recall refers to

Figure 1. File protocol specification



the proportion of sentences accepted by the language described by the true specification that are also accepted by the mined automaton. In the same work, a simulation framework is proposed to evaluate existing work on mining automaton-based specifications, which also help identify room for improvement.

Lo and Khoo (2006b) next propose an architecture using trace filtering and clustering to improve the quality of existing specification miners. The architecture pre-processes the traces by first filtering anomalies in the traces and then clustering them into similar groups. Each group is then fed separately to an existing specification mining algorithm to produce an automaton describing a partial specification of a system. These partial specifications are later merged into a unified automaton. It has been shown that the quality of the resultant specification mining algorithm after filtering and clustering are employed is better than before. In particular, in a case study on a Concurrent Versions System (CVS) application, the precision is doubled with a small reduction in recall.

Mariani, Papagiannakis and Pezzè (2007) use an improved automata learner (Mariani and Pezzè, 2005) and a Boolean expressions miner (Ernst, Cockrell, Griswold and Notkin, 2001) to generate regression tests of third party or Commercial off-the-shelf (COTS) components. Their algorithm first learns an automaton and a set of Boolean expressions and then converts them to regression tests to ensure the compatibility of third party components when used together.

Other work on mining automata includes: (Archaya, Xie, Pei & Xu, 2007; Reiss & Reinieris, 2001; Cook & Wolf 1998; etc.)

Mining Linear Temporal Logic Expressions

Linear Temporal Logic (LTL) is a formalism for specifying precise temporal requirements. It models time as a sequence of states where each state is an event from a fixed set. The full description of LTL can be found in (Huth & Ryan, 2004). Here, we focus on 3 temporal operators of LTL namely G, X and F. The operator X refers to the next state, *F* refers to the current or a future state, and *G* captures all future states (globally). In software there are many requirements expressible in LTL, for example:

1. Whenever (globally when) *resource.lock* is called, (from the next state onwards) finally *resource.unlock* is eventually called
   Or
   G(*resource.lock*->XF (*resource.unlock*))
2. Whenever (globally when) a correct pin is entered and user requested money and the balance is sufficient, (from the next state onwards) finally an ATM eventually dispenses money
   Or
   G(*correct_pin*->XG(*request_money*->XG(sufficient->XF(dispense))))

Yang, Evans, Bhardwaj, Bhat and Das (2006) mine significant two-event temporal logic expressions stating "whenever an event E_1 occurs eventually an event E_2 occurs" from program execution traces. An expression is significant if it satisfies a minimum threshold of ``satisfaction rate''. The algorithm is limited to mining two-event temporal logic expressions due to the exponential complexity involved in mining expressions of arbitrary sizes. To capture behaviors involving more than two events, they proposed a partial solution where previously mined two-event expressions are concatenated to form longer expressions. However, not all more-than-two event expressions can be mined this way, and superfluous rules that are not significant might be introduced in this process.

Lo, Khoo and Liu (2007b, 2008) extend this work by devising an algorithm to mine significant LTL expressions of arbitrary sizes. An expression is significant if it obeys minimum thresholds of support and confidence. A novel search space pruning strategy is employed to enable efficient mining of rules of arbitrary size.

Mining Frequent Patterns

Li and Zhou (2005) use a closed itemset mining algorithm by (Grahne & Zhu 2003) in their proposed algorithm called PR-Miner to recover elements of a program (function, variable, etc) that are often used together in a code. These associations among program elements can be composed as rules that reflect implicit programming rules in code.

El-Ramly, Stroulia and Sorenson (2002) propose a new pattern mining algorithm to mine frequent user-usage scenarios of a GUI based program composed of screens – these scenarios are termed as interaction patterns. Given a set of series of screen ids, frequent

Figure 2. A sample iterative pattern (of 32 events) mined from JBoss application server (note: diagram is read top to bottom, left to right)

Connection Set Up	TransactionManagerLocator.getInstance TransactionManagerLocator.locate TransactionManagerLocator.tryJNDI TransactionManagerLocator.usePrivateAPI	**Transaction Commit**	TxManager.commit TransactionImpl.commit TransactionImpl.beforePrepare TransactionImpl.checkIntegrity TransactionImpl.checkBeforeStatus TransactionImpl.endResources TransactionImpl.completeTransaction TransactionImpl.cancelTimeout TransactionImpl.doAfterCompletion TransactionImpl.instanceDone
TxManager Set Up	TxManager.begin XidFactory.newXid XidFactory.getNextId XidImpl.getTrulyGlobalId		
Transaction Set Up	TransactionImpl.associateCurrentThread TransactionImpl.getLocalId XidImpl.getLocalId LocalId.hashCode TransactionImpl.equals TransactionImpl.getLocalIdValue XidImpl.getLocalIdValue TransactionImpl.getLocalIdValue XidImpl.getLocalIdValue	**Transaction Disposal**	TxManager.releaseTransactionImpl TransactionImpl.getLocalId XidImpl.getLocalId LocalId.hashCode LocalId.equals

interaction patterns capturing common user interactions with the GUI are obtained.

Lo, Khoo and Liu (2007a) propose a new pattern mining algorithm to mine frequent patterns of program behavior from program execution traces —these patterns are termed as iterative patterns. Patterns can occur repeatedly within a program trace and across multiple traces. These patterns follow some constraints of Live Sequence Charts (LSC), which is one of the standards in software modeling community. An example of a specification mined from the transaction component of JBoss Application Server is shown in Figure 2. It specifies that a series of connection setup events is followed by transaction manager setup events, transaction setup events, transaction commit events and eventually transaction disposal events.

Mining Live Sequence Charts

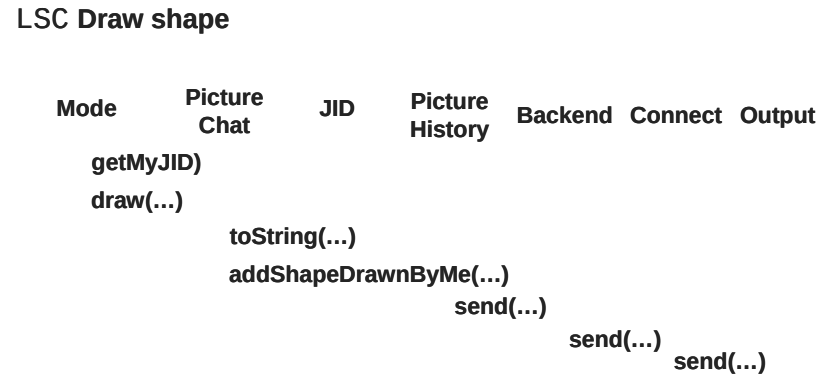
Lo, Maoz and Khoo (2007a, 2007b) extend the work on mining iterative pattern (Lo, Khoo & Liu, 2007a) and propose an algorithm to mine significant Live Sequence Charts (LSC) from program execution traces. A chart is significant if it obeys minimum thresholds of support and confidence. While iterative pattern follows some of the semantics of LSC they are not LSC. LSC is a formal version of UML sequence diagram with pre- and post- chart. An LSC specifies that when the behavior specified by the pre-chart occurs, the behavior specified by the post-chart will also occurs. Mined LSCs are also different from Linear Temporal Logic expressions mined in (Lo, Khoo & Liu, 2007b, Lo, Khoo & Liu, 2008) as the satisfactions of the pre- and post- charts conform to the semantics of LSC (Harel & Marelly 2003). Also for mining LSCs, an event in a trace is a triple (caller, callee, method signature). An example of a mined chart from Jeti instant messaging application (Jeti 2006) is shown in Figure 3. It describes a series of methods that are being called when a user of Jeti draws a line in the shared canvas with another communication party.

We have reviewed various approaches to mine different forms of software specifications. The specifications mined can aid user in understanding existing software and serve as input to other software engineering tasks, e.g., software verification, testing, etc.

FUTURE TRENDS

As program traces can be huge, there is a need to improve the efficiency of existing techniques further. Further industrial case studies are also needed to adapt existing techniques to the industry. A comparison of existing techniques will do well to help users to better understand which technique works best in a particular situation. There is much room for further theoretical and practical contributions to the domain of mining software specifications.

Figure 3. A sample LSC mined from Jeti messaginge application (boxes, arrows, dotted arrows and solid arrows correspond to classes, method calls, pre-chart and post-chart respectively)



CONCLUSION

Software is a ubiquitous component of our daily life. Documented software specifications are often missing, incomplete and outdated. This causes difficulties in understanding software systems. Program comprehension accounts for a significant proportion of total software cost. On another angle, programmers' reluctance and difficulty in writing formal specifications have been some barriers to the wide spread adoption of automatic program verification tools. Specification mining is one of the promising solutions to the above problems. It can provide candidate specifications to aid programmers in understanding existing programs. The specification can also be input to program verification tools and converted to run-time tests to aid program verification and bug detection. In the future, we look forward to more theoretical and technical contribution to this field, as well as more case studies and the availability of more open source tools.

REFERENCES

Ammons, G., Bodik, R., and Larus, J. (2002). Mining specifications. *Proceedings of the 29th Symposium on Principles of Programming Languages*, 4-16.

Archaya, M., Xie, T., J. Pei, and J. Xu (2007). Mining API patterns as partial orders from source code: from usage scenarios to specifications**.** *Proceedings of the 6th Joint Meeting of European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 25-34.

Canfora, G. and Cimitile, A. (2002). Software maintenance. *Handbook of Software Engineering and Knowledge Engineering (Volume 1*), 91-120, World Scientific.

Clarke, E.M., Grumberg, O. and Peled, D.A. (1999). *Model checking*. MIT Press.

Cook, J.E. and Wolf, A.L. (1998). Discovering models of software processes from event-based data. *ACM Transactions on Software Engineering and Methodology*, 7(3):215-249.

Deelstra, S., Sinnema, M. and Bosch, J. (2004). Experiences in software product families: Problems and issues during product derivation. *Proceedings of the 3rd Software Product Line Conference*, 165-182.

El-Ramly,.M, Stroulia, E., and Sorenson, P. (2002) From run-time behavior to usage scenarios: An interaction-pattern mining approach. *Proceedings of International Conference on Knowledge Discovery and Data Mining*, 315-324.

Erlikh, L. (2000). Leveraging legacy system dollars for e-business. *IEEE IT Pro*, 2(3), 17-23.

Ernst, M.D., Cockrell, J., Griswold, W.G., and Notkin, D. Dynamically discovering likely program invariants to support program evolution, *IEEE Transactions on Software Engineering*, 27(2), 99-123.

European Space Agency (ESA) and Centre National d'Etudes Spatiales (CNES) Independent Enquiry Board. (1996). ARIANE 5 – flight 501 failure: Report by the inquiry board. A copy at: http://www.ima.umn.edu/~arnold/disasters/ariane5rep.html.

Grahne, G. and Zhu, J. (2003). Efficiently using prefix-trees in mining frequent itemsets. *Proceedings of the 1st Workshop on Frequent Itemset Mining Implementation.*

Han, J. and Kamber, M. (2003). *Data mining: concepts and techniques.* Morgan Kaufmann.

Harel, D. and Marelly, R. (2003). *Come, let's play: Scenario-based programming using LSCs and the play-engine.* Springer-Verlag.

Holtzmann, G.J. (2002) The logic of bugs. (2002). *Proceedings of the 10th Symposium on Foundations of Software Engineering*, 81-87.

Hopcroft, J. E., Motwani, R. and Ullman, J.D. (2001). *Introduction to automata theory, languages, and computation.* Addison Wesley.

Huth, M. and Ryan, M. (2003). *Logic in computer science: Modeling and reasoning about systems.* Cambridge Press.

Jeti. Version 0.7.6 (Oct. 2006). http://jeti.sourceforge.net/.

Lehman, M. and Belady, L. (1985). *Program Evolution – Processes of Software Change*. Academic Press.

Lo, D., Khoo, S-C. (2006a). QUARK: Empirical assessment of automaton-based specification miners. *Proceedings of the 13th IEEE Working Conference on Reverse Engineering*, 51-60.

Lo, D., Khoo, S-C. (2006b). SMArTIC: Toward building an accurate, robust and scalable specification miner. *Proceedings of the 14th ACM SIGSOFT Symposium on Foundations of Software Engineering*, 265-275.

Lo, D., Khoo, S-C, Liu, C. (2007a). Efficient mining of iterative patterns for software specification discovery. *Proceedings of the 13th International Conference on Knowledge Discovery and Data Mining*, 460-469.

Lo, D., Khoo, S-C, Liu, C. (2007b). Mining temporal rules from program execution traces. *Proceedings of the 3rd International Workshop on Program Comprehension through Dynamic Analysis*, 24-28.

Lo, D., Khoo, S-C, Liu, C. (2008). Efficient mining of recurrent rules from a sequence database. *Proceedings of the 13th International Conference on Database Systems for Advance Applications*.

Lo, D., Maoz, S. and Khoo, S-C. (2007a). Mining modal scenarios from execution traces. *Companion to the 22nd Conference on Object-Oriented Programming, Systems, Languages, and Applications*, 777-778.

Lo, D., Maoz, S. and Khoo, S-C. (2007b). Mining modal scenario based specifications from execution traces of reactive systems. *Proceedings of the 22nd International Conference on Automated Software Engineering*, 465-468.

Lo, D., Khoo, S-C. (2007). Software specification discovery: A new data mining approach. *NSF Symposium on Next Generation Data Mining*.

Mariani, L., Papagiannakis and S., Pezzè, (2007). Compatibility and regression testing of COTS-component-based software. *Proceedings of the 29th International Conference on Sofware Engineering*, 85-95.

Mariani, L. and S., Pezzè, (2005). Behavior capture and test: Automated analysis of component integration. *Proceedings of the 10th International Conference on Engineering of Complex Computer Systems*, 292-301.

Raman, A.V. and Patrick, J.D. (1997). The sk-strings method for inferring PFSA. *Proceedings of the Workshop on Automata Induction, Grammatical Inference and Language Acquisition*.

Reiss, S.P. and Renieris, M. (2001). Encoding program executions. *Proceedings of the International Conference on Software Engineering*, 221-230.

Standish, T. (1984). An essay on software reuse. IEEE *Transactions on Software Engineering*, 5(10): 494-497.

US Bureau of Economic Analysis (BEA). (Sept 2007) BEA: News release: Gross domestic product. Online at: http://www.bea.gov/newsreleases/national/gdp/gdpnewsrelease.htm.

US National Institute of Standards and Technology (NIST). (2002). Software errors cost U.S. economy $59.5 billion annually. Online at: http://www.nist.gov/public affairs/releases/n02-10.htm

US Government Accountability Office (GAO). (1992). GAO report: Patriot missile defense – software problem led to system failure at Dhahran, Saudi. A copy at: http://www.fas.org/spp/starwars/gao/im92026.htm.

Yang, J., Evans, D., Bhardwaj, D., Bhat, T. and Das, M. (2006). Perracotta: Mining temporal API rules from imperfect traces. *Proceedings of the 28th International Conference on Software Engineering*, 282-291.

KEY TERMS

Automaton: A labeled transition system with start and end nodes describing a language. A path from the start to an end node corresponds to a sentence in the language.

Linear Temporal Logic: Formalism commonly used to describe temporal requirements precisely. There are a few basic operations given with symbols G, X, F, U, W, R corresponding to English language terms 'Globally', 'neXt', 'Finally', 'Until', 'Weak-until' and 'Release'.

Live Sequence Charts: A formal version of UML sequence diagram. It is composed of a pre- and post-chart. The pre-chart describes a condition which if satisfied entails that the behavior described in the post-chart will occur.

Program Comprehension: A process of understanding a piece of software.

Program Instrumentation: Simply put, it is a process of inserting `print' statements to an existing program such that by running the instrumented program, it will produce a trace file reflecting the behavior of the program when the program is run.

Program Testing: A process find defects in a piece of software by running a set of test cases.

Program Trace: A series of events where each event can correspond to a statement that is being executed, a function that is being called etc., depending on the abstraction level considered in producing the trace.

Program Verification: A process to ensure that software is always correct no matter what input is given with respect to some properties, e.g., whenever a resource is locked for usage, it is eventually released.

Software Maintenance: A process of incorporating changes to existing software, e.g., bug fixes, feature additions, etc., while ensuring the resultant software works well.

Specification Mining or Specification Discovery: A process for automated extraction of software specification from program artifacts. We use artifacts in a liberal sense to include program traces, code, repository, data, etc.

Software Specification: A description on how a piece of software is supposed to behave.

Mining the Internet for Concepts

Ramon F. Brena
Tecnológico de Monterrey, Mexico

Ana Maguitman
Universidad Nacional del Sur, Argentina

Eduardo H. Ramirez
Tecnológico de Monterrey, Mexico

INTRODUCTION

The Internet has made available a big number of information services, such as file sharing, electronic mail, online chat, telephony and file transfer. However, services that provide effective access to Web pages, such as Google, are the ones that most contributed to the popularization and success of the World Wide Web and the Internet. Pages published at the World Wide Web belong to many different topic areas, such as music, fishing, travel, etc. Some organizations have tried to organize pages in a predefined classification, and have manually built large directories of topics (e.g. *Dmoz* or the *Yahoo!* directory). But given the huge size and the dynamic nature of the Web, keeping track of pages and their topic manually is a daunting task. There is also the problem of agreeing on a standard classification, and this has proved to be a formidable problem, as different individuals and organizations tend to classify things differently. Another option is to rely on automatic tools that mine the Web for "topics" or "concepts" related to online documents. This approach is indeed more scalable than the manual one. However, automatically classifying documents in topics is a major research challenge. This is because the document keywords alone seem to be insufficient to directly convey the meaning of the document to an autonomous system. In some cases, the main difficulty is due to the ambiguity of the terms encountered in the document. Even if the ambiguity problems were solved there is still no guarantee that the vocabulary used to describe the document will match that used by the autonomous system to guide its search.

Central to automatic approaches is the notion of "semantic context", which loosely means the subject or topic where a task like searching is embedded. Of course, we need a way to computationally represent this notion of context, and one possibility is to see context as a collection of interrelated terms in the sense that they appear together in a number of related pages (Ramirez & Brena, 2006). For instance, the word "Java" appears together with "roasted" when talking about coffee, but appears more frequently with "code" when talking about a programming language. Semantic contexts allow performing searches on the Web at the concept level, rather than at the more basic keyword level. In this chapter we present recent advances in automated approaches in web concept mining, emphasizing our own work about mining the Web for semantic contexts.

BACKGROUND

The notion of semantic contexts is closely linked to that of *semantic similarity*. Two documents are semantically similar if they belong to the same topic or to similar topics. Likewise, two words are semantically similar if they represent similar concepts.

The study of semantic similarity between documents and terms has long been an integral part of information retrieval and machine learning, and there is extensive literature on measuring the semantic similarity between entities (Resnik, 1995; Lin, 1998; Hatzivassiloglou et al. 1999; Landauer et al. 1997; Turney, 2001; Maguitman et al. 2005; Ramirez & Brena, 2006; Sahami et al. 2006; Bollegala, 2007). These methods can be roughly classified into two major categories: knowledge-based and corpus-based approaches.

Knowledge-based approaches rely on the use of predefined directories or ontologies to identify semantic relations. Measures of semantic similarity between entities take as a starting point the structure of a directory or ontology where the entities have been previously classified. Examples of such ontologies include

WordNet for the case of terms, and *Dmoz* or the *Yahoo! Directory* for the case of document. Ontologies are a special kind of network and early proposals to estimate semantic similarity have used path distances between the nodes in the network representation (Rada et al. 1989). Some successors have looked into the notion of information content (Resnik, 1995; Lin, 1998) or have combined both distance and information content (Jiang & Conrath, 1997) to assess semantic similarity. In an information theoretic approach, the semantic similarity between two entities is related to their commonality and to their differences. Given a set of entities in a hierarchical taxonomy, the commonality of two entities can be estimated by the extent to which they share information, indicated by the most specific class in the hierarchy that subsumes both. Once this common classification is identified, the meaning shared by two entities can be measured by the amount of information needed to state the commonality of the two objects. Generalizations of the information theoretic notion of semantic similarity for the case of general ontologies (i.e., taxonomies that include both hierarchical and non-hierarchical components) have been proposed by Maguitman et al. (2005).

Keeping these ontologies up-to-date is expensive. For example, the semantic similarity between terms changes across different contexts. Take for instance the term java, which is frequently associated with the java programming language among computer scientist. However, this sense of java is not the only one possible as the term may be referring to the java coffee, the java island or the java Russian cigarettes, among other possibilities. New words are constantly being created as well as new senses are assigned to existing words. As a consequence, the use of knowledge-based approaches has disadvantages because they require that the ontologies be manually maintained.

Corpus-based approaches, on the other hand, can help to automate the process of keeping the ontology up-to-date. They are based on information exclusively derived from large corpora (such as the World Wide Web). A well-known approach of corpora-based method is latent semantic analysis (Landauer et al. 1997), which applies singular value decomposition to reduce the dimensions of the term-document space, harvesting the latent relations existing between documents and between terms in large text corpora. Less computationally expensive techniques are based on mapping documents to a kernel space where documents that do not share any term can still be close to each other (Cristianini et. al 2001; Liu et al. 2004). Another corpus-based technique that has been applied to estimate semantic similarity is PMI-IR (Turney, 2001). This information retrieval method is based on pointwise mutual information, which measures the strength of association between two elements (e.g., terms) by contrasting their observed frequency against their expected frequency.

In general, automatic methods to compute similarity between texts have applications in many information retrieval related areas, including natural language processing and image retrieval from the Web, where the text surrounding the image can be automatically augmented to convey a better sense of the topic of the image. Automatic methods to identify the topic of a piece of text have also been used in text summarization (Erkan and Radev 2004), text categorization (Ko et al. 2004), word sense disambiguation (Schutze 1998), evaluation of text coherence (Lapata and Barzilay 2005) and automatic translation (Liu and Zong 2004), but most of them use human-defined categories for topics, becoming thus prone to the problems we mentioned before, like disagreement, difficulty to update, etc.

MAIN FOCUS

We discuss two completely corpus-based methods proposed by the authors that can be used to identify semantic contexts and applied to mine the Web for concepts. The first is based on the notion of k-core and the second is based on the use of incremental methods.

K-Core Method

In the k-core method, the main idea is to find groups of *k* keywords (for instance, 4 keywords) that appear together in a big number of pages. The basic assumption is that words with related meanings appear together more often than unrelated words. The number of co-occurrences is readily obtained using the current indexing-based search methods; for instance, *Google* search includes this number in every single search results page (upper-right corner). So, co-occurrences based methods could be extremely efficient and well integrated with internet search technology.

A "Semantic Context" is defined in this method as the relative weights of keywords in a given topic. For instance, when talking about coffee, the word sugar is

very important, but it is irrelevant when the subject is automobile engines. Formally, a Semantic Context is a function from keywords to numbers from 0 to 1, 1 meaning that the given word is of utmost importance and 0 of no importance. Semantic Contexts represent conceptual topics, through the relative importance of the words used in documents belonging to that topic in a given corpus.

Now, Semantic Contexts require to locate a group of documents all of them "talking" about a certain topic, and then standard TF-IDF measures are calculated for each keyword. Those documents could of course be compiled manually, but in (Ramirez & Brena, 2006) this is done using a device called "k-cores". As explained before, k-cores are groups of *k* words that appear together in a big number of documents in the corpus, indicating thus that they are semantically linked. In the k-core method a value called "force" is calculated for each possible group of *k* words; the force is a kind of measure of how good are the given words for describing a topic. A formula has been proposed for calculating force, which takes into account the joint frequency of the words in the candidate k-core; for the case of two words k_1 and k_2 the force would be:

$$F(k_1,k_2)/(F(k_1,\sim k_2)+F(k_2,\sim k_1))$$

where $F(k_1,k_2)$ gives the joint document frequency of k_1 and k_2, and "~" represents negation, so $F(k_2,\sim k_1)$ represents the quantity of documents having k_2 but not k_1. For more than 2 words, the formula gets more complicated, but the idea is the same.

Finally, a k-core is just a group of group of *k* words of maximal force, meaning that any single-word replacement would decrease the force.

Once k-cores are calculated, they are used as conjunctive queries against the corpus, obtaining sets of documents belonging to the topic of the corresponding k-core; from those sets of documents, the Semantic Contexts are calculated as explained above. Then, distances between two Semantic Contexts are easily calculated taking them as vectors.

Experimental results (Ramirez & Brena, 2006) show that: 1) the method is able to find k-cores for all the topics in a given corpus, 2) Distances between the calculated k-cores actually correspond to intuitive closeness of topics.

In (Ramirez & Brena, 2006) are presented specific algorithms, as well as applications of this method, such as focusing of Internet searches, helping thus the user to disambiguate an initial search so that only results from a chosen topic are filtered.

Further work is being done in the k-core method in order to increase the efficiency for calculating all the k-cores for a large corpus.

Incremental Methods

When seeking for material semantically related to a document, it is natural to form queries using the most descriptive terms of the documents. We have developed an incremental method that exploits the search context to go beyond the vocabulary of the initial document. The method starts with a small number of terms randomly selected from the document under analysis and uses the full document as the initial search context. Using these terms, a set of queries are built and submitted to a search engine. After the first set of results has been obtained, the search context is used to refine/extend the set of terms used for the context description. Terms that appear *often* in search results similar to the search context tend to be good "descriptors" of the thematic context. On the other hand, terms that tend to occur *only* in results similar to the search context can serve as "discriminators".

Intuitively, we can characterize topic descriptors and discriminators as follows:

1. Terms are good topic descriptors if they answer the question "What is this topic about?"
2. Terms are good topic discriminators if they answer the question "What are good query terms to access similar information?"

In this approach *topic* descriptors and discriminators are considered higher-order notions and distinguished from the more basic notions of *document* descriptors and discriminators.

Formally, given a collection of *m* documents and *n* terms we can build a $m \times n$ matrix *H*, such that $H[i,j]=k$, where *k* is the number of occurrences of term t_j in document d_i. We define discriminating power of a term in a document as a function $\delta: \{t_0,\ldots,t_{n-1}\} \times \{d_0,\ldots,d_{m-1}\} \rightarrow [0,1]$:

$$\delta(t_i, d_j) = \frac{\mathrm{sgn}(H^T[i,j])}{\sqrt{\sum_{k=0}^{m-1} \mathrm{sgn}(H^T[i,k])}}$$

Analogously, we define descriptive power of a term in a document as a function $\lambda: \{d_0,\ldots,d_{m-1}\} \times \{t_0,\ldots,t_{n-1}\} \rightarrow [0,1]$:

$$\lambda(d_i, t_j) = \frac{H[i,j]}{\sqrt{\sum_{k=0}^{n-1} (H[i,k])^2}}$$

These simple notions of document descriptors and discriminators share some insight with standard IR proposals. However here we are interested in a topic-dependant definition of topic descriptors and discriminators. We formally define the descriptive power of a term in the topic of a document as a function $\Lambda: \{d_0,\ldots,d_{m-1}\} \times \{t_0,\ldots,t_{n-1}\} \rightarrow [0,1]$ calculated as follows:

$$\Lambda(d_i, t_j) = \begin{cases} 0 & \text{if } \sum_{k=0,\ k\neq i}^{m-1} \sigma(d_i, d_k) = 0 \\ \dfrac{\sum_{k=0,\ k\neq i}^{m-1} (\sigma(d_i, d_k) \cdot \lambda(d_k, t_j)^2)}{\sum_{k=0,\ k\neq i}^{m-1} \sigma(d_i, d_k)} & \text{otherwise} \end{cases}$$

where $\sigma(d_i,d_j)$ stands for some similarity measure between documents d_i and d_j. Thus the discriminating power of term t_i in the topic of document d_j is an average of the similarity of d_j to other documents discriminated by t_i. Indeed, the descriptive power of a term in a topic is defined using the simpler notions of document similarity and document descriptors: The topic descriptive power of a term *t* in the topic of a document *d* is a measure of the quality of *t* as a descriptor of documents similar to *d*.

Analogously, the discriminating power of a term *t* in the topic of a document *d* is quantified as the average of the similarity of *d* to other documents discriminated by *t* and is formally defined as a function $\Delta: \{t_0,\ldots,t_{n-1}\} \times \{d_0,\ldots,d_{m-1}\} \rightarrow [0,1]$ calculated as follows:

$$\Delta(t_i, d_j) = \sum_{k=0,\ k\neq j}^{m-1} (\sigma(d_k, d_j) \cdot \partial(t_i, d_k)^2)$$

Thus the discriminating power of term t_i in the topic of document d_j is an average of the similarity of d_j to other documents discriminated by t_i.

When used as query terms, topic descriptors are useful to improve recall because they occur in many similar documents. Topic discriminators, on the other hand, help restrict the set of search results to mostly similar material and therefore can help achieve high precision. A formal characterization of topic descriptors and discriminators as well as an evaluation of their usefulness as query terms can be found in (Maguitman et al. 2004).

Our incremental methods identify topic descriptors and topic discriminators by analyzing the terms in retrieved documents. Consequently, they are not restricted to terms occurring in the originating search context, and if novel terms have high descriptive or discriminating power, they expand the vocabulary found in the initial document. In this incremental search process, the generation of second-round and subsequent queries can significantly benefit from a search context refined by the addition of good topic descriptors and discriminators.

Advantages and Disadvantages

The two methods just described are quite similar, because they aim to automatically find relevant terms for characterizing topics. They differ in that k-cores do not distinguish between descriptors and discriminators, but enforce a notion of joint occurrence as a topic definer.

As we previously mentioned, the main quality of both of these two methods is that they compute topics in an automated way, departing thus from many known methods that start with human-defined sets of topics. We insist in saying that automatically computed topics are less prone than human-made ones to arbitrary classifications and thus disagreements, and they can be more comprehensive and easy to update.

We currently do not know if there are disadvantages in automatic methods. It is possible that in real-world applications human and machine-made topic classifications could coexist.

FUTURE TRENDS

Semantic context-based methods to mine the Web for concepts are gradually becoming a key component

of today's Internet infrastructure. The techniques discussed here can be applied to any domain for which it is possible to generate term-based characterizations of a topic. We anticipate that a number of information services will benefit from these methods. The areas of application could include topic-based and task-based search, semantic web navigation, automated translations, and support for knowledge management, among others. To illustrate the automated translation application, consider the situation where a

CONCLUSION

The context in which an Internet task such as a search is done is of huge practical importance to users. Traditional IR techniques fall short for accessing Web resources in many highly context-dependent situations. One very important form of context is the topic or concept to which the task is related. But collecting and organizing topics that serve as contexts is not an easy task. In this chapter we have reviewed techniques that allow for dynamic identification of useful context-specific material, some of them based on the notion of semantic similarity.

Methods that take advantage of the information available in large corpora, such as the Web, to generate a semantic context are particularly useful to go beyond the predefined set of terms in a document. In contrast to knowledge-based approaches, corpus-based approaches, as the two ones described in this chapter, do not require human editors to manually classify documents into topics or to associate term with concepts. They actually aim to mining a given corpus for finding its topics, which can be used as contexts. The advantages of these methods have been cited before: comprehensiveness, objectivity, and capability to adjust to the dynamic nature of the Web and human knowledge in general.

REFERENCES

Bollegala, D. Measuring Semantic Similarity Between Words using Web Search Engines. *In Proceedings of the Sixteenth International Conference on World Wide Web (WWW)*. 757-766. ACM Press, New York, NY

Cristianini, N., Shawe-Taylor, J. and Lodhi, H. (2001). Latent semantic kernels. In Proceedings the Eighteenth International Conference on Machine Learning (ICML 01), 66-73. Morgan Kaufmann Publishers, San Francisco, US.

Erkan, G. and Radev, D.R. (2004). LexRank: Graph-Based Lexical Centrality As Salience in Text Summarization. J. Artificial Intelligence Research 22, 457-479.

Hatzivassiloglou, V., Klavans, J., and Eskin, E. (1999). Detecting Text Similarity over Short Passages: Exploring Linguistic Feature Combinations via Machine Learning. *In Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP 99)*, 203-212.

Jiang, J. J. and Conrath, D.W. (1997). Semantic Similarity based on Corpus Statistics and Lexical Taxonomy. *Proceedings of International Conference Research on Computational Linguistics, (ROCLING X)*, 19-33.

Ko, Y., Park, J., and Seo, J. (2004). Improving Text Categorization Using the Importance of Sentences. Information Processing and Management 40, 65-79.

Landauer, T.K. and Dumais, S.T. (1997). A Solution to Plato's Problem: The Latent Semantic Analysis Theory of the Acquisition, Induction, and Representation of Knowledge. Psychological Review 104(2), 211-240.

Lapata M. and Barzilay R. (2005). Automatic evaluation of text coherence: Models and representations. Proceedings of IJCAI-05, 2005, Edinburgh, Scotland, UK.

Lin, D. (1998). An Information-theoretic Definition of Similarity. *In Proceedings of the Fifteenth International Conference on Machine Learning (ICML 98)*. 296-304. Morgan Kaufmann, San Francisco, CA.

Liu, N., Zhang, B., Yan, J., Yang, Q., Yan, A., Chen, Z., Bai, F., and. Ma, W.-Y. (2004). Learning similarity measures in non-orthogonal space. *Proceedings of the Thirteenth Conference on Information and Knowledge Management (CIKM 04)*, 334-341. New York: ACM Press.

Liu, Y. and Zong, C.Q. (2004). Example-Based Chinese-English MT. In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics 1-7, 6093-6096.

Maguitman, A.; Leake, D.; Reichherzer, T.; and Menczer, F. (2004). Dynamic extraction of topic descriptors and discriminators: Towards automatic context-based topic search. In *Proceedings of the Thirteenth Conference on Information and Knowledge Management (CIKM 04)*, 463-472. New York: ACM Press.

Maguitman, A., Menczer, F., Roinestad, H., and Vespignani, A. (2005). Algorithmic Detection of Semantic Similarity. *In Proceedings 14th Int'l World Wide Web Conference (WWW 05)* 107 - 116, New York: ACM Press.

Rada, R., Mili, H., Bicknell, E., Blettner, M. (1989). Development and Application of a Metric on Semantic Nets. IEEE Transactions on Systems, Man and Cybernetics. 19(1), 17–30. IEEE Press.

Ramirez E. and Brena R. (2006). Semantic Contexts in the Internet. *Proceedings of the Forth Latin American Web Congress. (LA-Web'06).* 74-81, IEEE Press.

Resnik, P. (1995). Using Information Content to Evaluate Semantic Similarity in a Taxonomy. *In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI 95). 448-453.*

Sahami M. and Heilman, T. (2006). A Web-Based Kernel Function for Measuring the Similarity of Short Text Snippets. *In Proceedings of the Fifteenth International World Wide Web Conference (WWW 06),* 377-386. ACM Press, New York, NY.

Schutze, H. (1998). Automatic Word Sense Discrimination. Computational Linguistics 24, 1, 97–124.

Turney, P. (2001). Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL. *Proceedings of the Twelfth European Conference on Machine Learning (ECML 01)*, 491-502, Morgan Kaufmann, Freiburg, Germany.

KEY TERMS

Ambiguity: A same phrase having more than one meaning, like "flying planes", that could be a noun (planes that fly) or verb (the activity of flying the planes).

Context: The set of conditions embedding and influencing a given task. In this chapter we focus mainly in contexts understood as topics, subjects or concepts, like music, engines, fishing, etc.

Corpus, Corpora: The set of all available documents.

Disambiguation: The action of selecting one particular meaning among several possible ones in an ambiguous expression (see "ambiguity" above).

Hierarchical taxonomy: Tree-like classification of categories in subcategories, then them in subcategories, etc., like the living beings in animals and plants, etc.

Information retrieval: Field studying the techniques and methods for getting relevant information in an efficient way.

Information Theory: Discipline aiming to measure the quantity of information transmitted or stored.

Keywords: Nouns, verbs and adjectives, excluding thus non-relevant words for purposes of word content, like "in", "or", etc.

Machine Learning: The study of methods that allow computers to improve their performance using past experience.

Ontologies: Definition of classes of objects, attributes, properties and relations to other objects, expressed in Semantic Web markup languages such as "OWL".

Semantic Similarity: Closeness in meaning, using some metric for measuring the distance between words, documents, etc.

Web Mining: The search for patterns in the Web using data mining techniques.

Model Assessment with ROC Curves

Lutz Hamel
University of Rhode Island, USA

INTRODUCTION

Classification models and in particular binary classification models are ubiquitous in many branches of science and business. Consider, for example, classification models in bioinformatics that classify catalytic protein structures as being in an active or inactive conformation. As an example from the field of medical informatics we might consider a classification model that, given the parameters of a tumor, will classify it as malignant or benign. Finally, a classification model in a bank might be used to tell the difference between a legal and a fraudulent transaction.

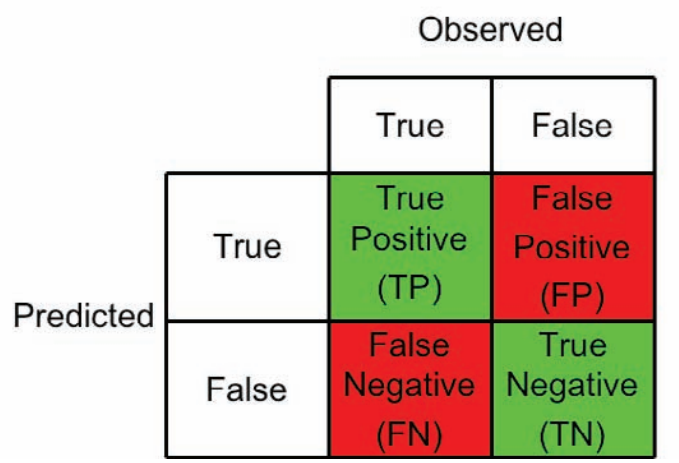
Central to constructing, deploying, and using classification models is the question of model performance assessment (Hastie, Tibshirani, & Friedman, 2001). Traditionally this is accomplished by using metrics derived from the confusion matrix or contingency table. However, it has been recognized that (a) a scalar is a poor summary for the performance of a model in particular when deploying non-parametric models such as artificial neural networks or decision trees (Provost, Fawcett, & Kohavi, 1998) and (b) some performance metrics derived from the confusion matrix are sensitive to data anomalies such as class skew (Fawcett & Flach, 2005). Recently it has been observed that Receiver Operating Characteristic (ROC) curves visually convey the same information as the confusion matrix in a much more intuitive and robust fashion (Swets, Dawes, & Monahan, 2000).

Here we take a look at model performance metrics derived from the confusion matrix. We highlight their shortcomings and illustrate how ROC curves can be deployed for model assessment in order to provide a much deeper and perhaps more intuitive analysis of the models. We also briefly address the problem of model selection.

BACKGROUND

A binary classification model classifies each instance into one of two classes; say a *true* and a *false* class. This gives rise to four possible classifications for each

Figure 1. Format of a confusion matrix



instance: a true positive, a true negative, a false positive, or a false negative. This situation can be depicted as a confusion matrix (also called contingency table) given in Figure 1. The confusion matrix juxtaposes the observed classifications for a phenomenon (columns) with the predicted classifications of a model (rows). In Figure 1, the classifications that lie along the major diagonal of the table are the correct classifications, that is, the true positives and the true negatives. The other fields signify model errors. For a perfect model we would only see the true positive and true negative fields filled out, the other fields would be set to zero. It is common to call true positives *hits*, true negatives *correct rejections*, false positive *false alarms*, and false negatives *misses*.

A number of model performance metrics can be derived from the confusion matrix. Perhaps, the most common metric is *accuracy* defined by the following formula:

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Other performance metrics include *precision* and *recall* defined as follows:

$$\text{precision} = \frac{TP}{TP + FP},$$

$$\text{recall} = \frac{TP}{TP + FN}$$

Note, that when we apply a model to a test dataset we obtain only one scalar value for each performance metric. Figure 2 shows two confusion matrices of one particular classification model built on the ringnorm data by Breiman (Breiman, 1996). Part (a) shows the classification model being applied to the original test data that consists of 7400 instances roughly split evenly between two classes. The model commits some significant errors and has an accuracy of 77%. In part (b) the model is applied to the same data but in this case the negative class was sampled down by a factor of ten introducing class skew in the data. We see that in this case the confusion matrix reports accuracy and precision values that are much higher than in the previous case. The recall did not change, since we did not change anything in the data with respect to the 'true' class. We can conclude that the perceived quality of a model highly depends on the choice of the test data. In the next section we show that ROC curves are not

Figure 2. Confusion matrices with performance metrics. (a) confusion matrix of a model applied to the original test dataset, (b) confusion matrix of the same model applied to the same test data where the negative class was sampled down by a factor of ten

so dependent on the precise choice of test data, at least with respect to class skew.

MAIN FOCUS OF CHAPTER

ROC Curves: The Basics

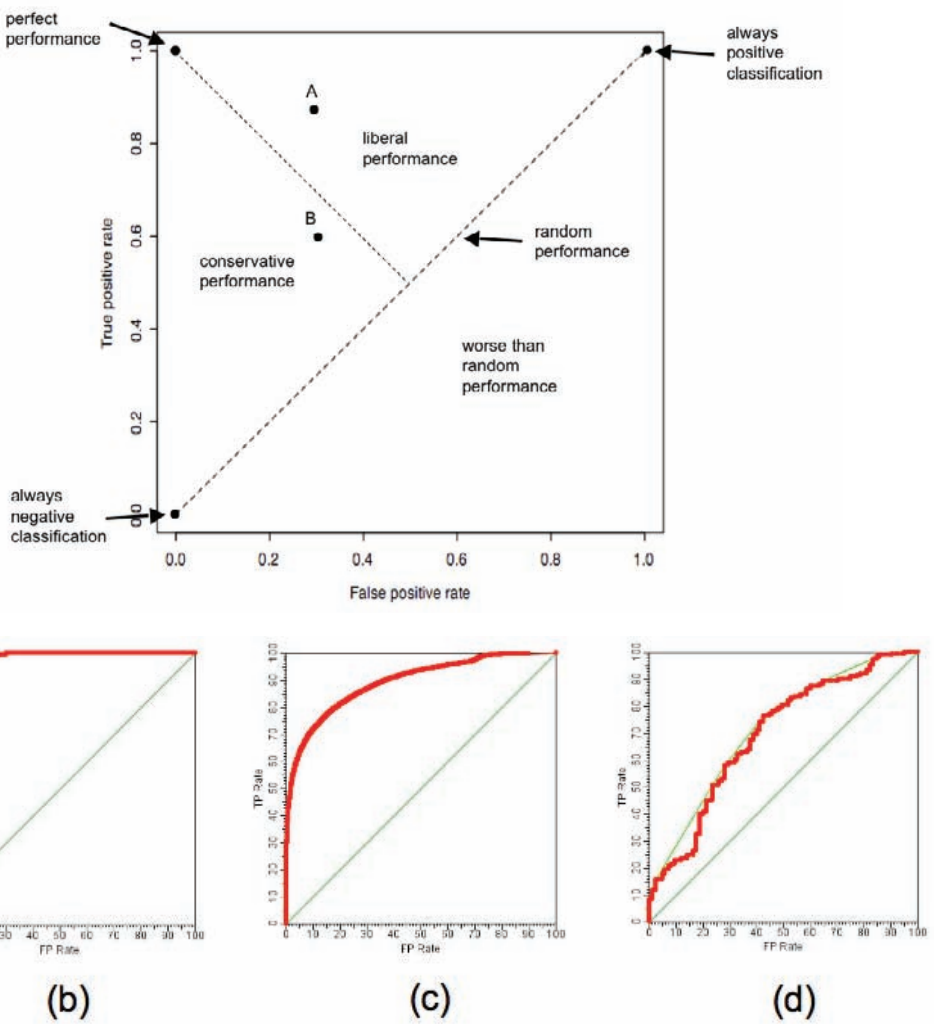
ROC curves are two-dimensional graphs that visually depict the performance and performance trade-off of a classification model (Fawcett, 2004; P. Flach, Blockeel, Ferri, Hernandez-Orallo, & Struyf, 2003; P. Flach, 2004; P. A. Flach, 2003). ROC curves were originally designed as tools in communication theory to visually determine optimal operating points for signal discriminators (Egan, 1975).

We need to introduce two new performance metrics in order to construct ROC curves (we define them here in terms of the confusion matrix), the *true positive rate* (tpr) and the *false positive rate* (fpr):

$$\text{true positive rate} = \frac{TP}{TP + FN} = \text{recall},$$

$$\text{false postive rate} = \frac{FP}{TN + FP}.$$

Figure 3. ROC curves: (a) regions of a ROC graph (b) an almost perfect classifier (c) a reasonable classifier (d) a poor classifier [1]



ROC graphs are constructed by plotting the true positive rate against the false positive rate (see Figure 3(a)). We can identify a number of regions of interest in a ROC graph. The diagonal line from the bottom left corner to the top right corner denotes random classifier performance, that is, a classification model mapped onto this line produces as many false positive responses as it produces true positive responses. To the left bottom of the random performance line we have the conservative performance region. Classifiers in this region commit few false positive errors. In the extreme case, denoted by point in the bottom left corner, a conservative classification model will classify all instances as negative. In this way it will not commit any false positives but it will also not produce any true positives. The region of classifiers with liberal performance occupies the top of the graph. These classifiers have a good true positive rate but also commit substantial numbers of false positive errors. Again, in the extreme case denoted by the point in the top right corner, we have classification models that classify every instance as positive. In that way, the classifier will not miss any true positives but it will also commit a very large number of false positives. Classifiers that fall in the region to the right of the random performance line have a performance worse than random performance, that is, they consistently produce more false positive responses than true positive responses. However, because ROC graphs are symmetric along the random performance line, inverting the responses of a classifier in the "worse than random performance" region will turn it into a well performing classifier in one of the regions above the random performance line. Finally, the point in the top left corner denotes perfect classification: 100% true positive rate and 0% false positive rate.

The point marked with A is the classifier from the previous section with a tpr = 0.90 and a fpr = 0.35. Note, that the classifier is mapped to the same point in the ROC graph regardless whether we use the original test set or the test set with the sampled down negative class illustrating the fact that ROC graphs are not sensitive to class skew.

Classifiers mapped onto a ROC graph can be ranked according to their distance to the 'perfect performance' point. In Figure 3(a) we would consider classifier A to be superior to a hypothetical classifier B because A is closer to the top left corner.

The true power of ROC curves, however, comes from the fact that they characterize the performance of a classification model as a curve rather than a single point on the ROC graph. In addition, Figure 3 shows some typical examples of ROC curves. Part (b) depicts the ROC curve of an almost perfect classifier where the performance curve almost touches the 'perfect performance' point in the top left corner. Part (c) and part (d) depict ROC curves of inferior classifiers. At this level the curves provide a convenient visual representation of the performance of various models where it is easy to spot optimal versus sub-optimal models.

ROC Curve Construction

In order to interpret ROC curves in more detail we need to understand how they are constructed. Fundamental to the construction of ROC curves is the notion of instance ranking or prediction confidence value. ROC curves can be directly computed for any classification model that attaches a probability, confidence value, or ranking to each prediction. Many models produce such rankings as part of their algorithm (e.g. Naïve Bayes (Mitchell, 1997), Artificial Neural Networks (Bishop, 1995), Support Vector Machines (Cristianini & Shawe-Taylor, 2000)). Techniques exist that compute an instance ranking for classification models that typically do not produce such rankings, i.e., decision trees (Breiman, Friedman, Olshen, & Stone, 1984). The instance ranking is used by the ROC algorithm to sweep through different decision thresholds from the maximum to the minimum ranking value in predetermined increments. The ranking values are typically normalized to values between 0 and 1 (as an aside, the default decision threshold for most classifiers is set to .5 if the ranking value expresses the actual probability value of the instance being classified as true). At each threshold increment, the performance of the model is computed in terms of the true positive and false positive rates and plotted. This traces a curve from left to right (maximum ranking to minimum ranking) in the ROC graph. That means that the left part of the curve represents the behavior of the model under high decision thresholds (conservative) and the right part of the curve represents the behavior of the model under lower decision thresholds (liberal).

The following algorithm[2] makes this construction a little bit more concrete:

Function Draw-ROC

Inputs:
D: test set
p(i): ranking of instance i in D, indicates the probability or confidence that the instance i is positive, normalized to [0,1]
P: set of observed positive instances in D, where $P \subseteq D$
N: set of observed negative instances in D, where $N \subseteq D$

```
for threshold = 1 to 0 by -.01 do
  FP ⇐ 0
  TP ⇐ 0
  for i ∈ D do
      if p(i) ≥ threshold then
              if i  ∈ P then
                      TP ⇐ TP + 1
              else
                      FP ⇐ FP + 1
              endif
      endif
  endfor
  tpr ⇐ TP/#P
  fpr ⇐FP/#N
  Add point (tpr, fpr) to ROC curve
endfor
```

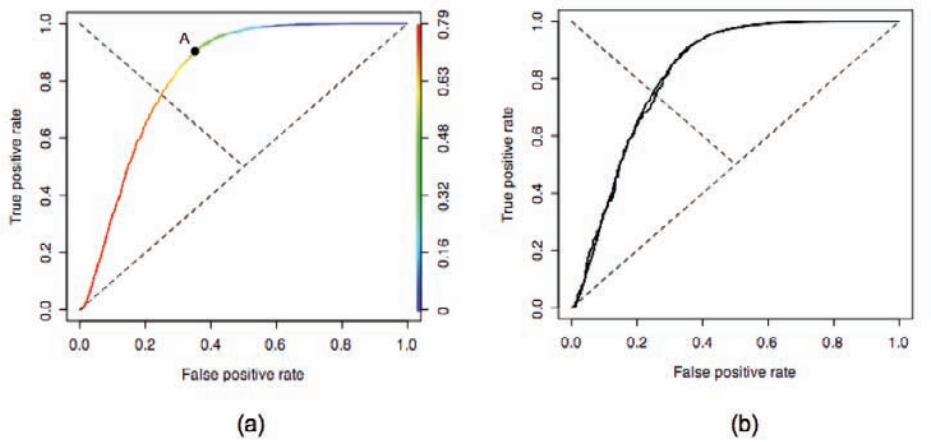
Notice how this algorithm sweeps through the range of thresholds from high to low and measures the number of mistakes the classifier makes at each threshold level. This gives rise to the tpr and fpr at each threshold level. This in turn can be interpreted as a point on the ROC curve.

Figure 4(a) shows the ROC curve of the classifier from Figure 2 with the decision thresholds annotated in color. From this we can see that the optimal decision threshold for this model (maximum tpr, minimum fpr, also called optimal operating point) occurs at a threshold of .35 in the green region representing a tpr = .95 and an fpr = .45. As we would expect from our confusion matrix analysis, we can observe that it is a reasonable classifier. We can also observe that it is a liberal classifier in that the optimal decision threshold of the curve lies in the liberal region of the ROC graph. It is also interesting to observe that the performance given by the confusion matrix maps to a suboptimal point on the curve (given as 'A' on the curve). This is due to the fact that the classification reported in the confusion matrix is based on the default decision threshold value of .5 instead of the optimal threshold value of .35.

In Figure 4(b) we can see two ROC curves for the same classifier as in part (a), one is based on the original test data and the other one is based on the skewed test data. Both curves are virtually identical illustrating that ROC curves are not sensitive to class skew.

Returning to Figure 3 above, we can now interpret these curves a little bit more carefully. In part (b) we see that the model only begins to commit false positive errors after it has almost reached a true positive rate of 100%. This means that at this point the decision

Figure 4. ROC curves of the classifier given in Fig. 2. (a) ROC curve with decision threshold values, (b) ROC curves of the classifier evaluated against original test data and the down sampled data



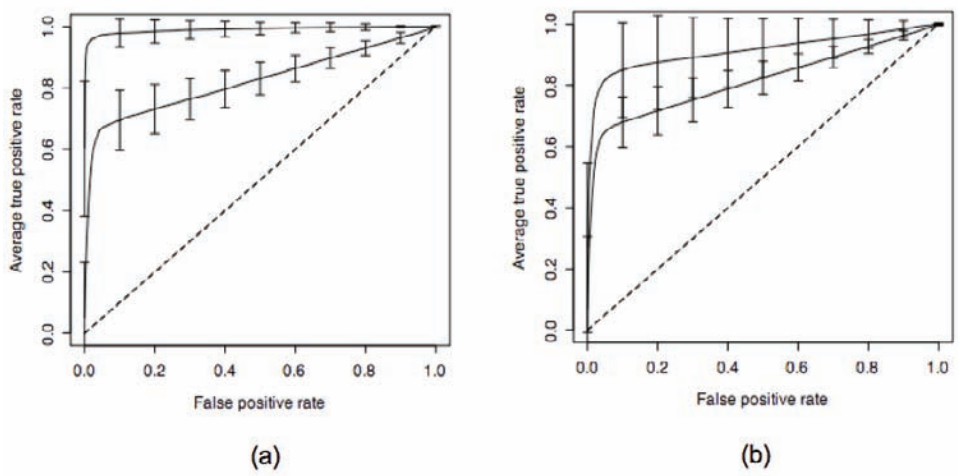
threshold has been lowered to a point that observed, negative instances are classified as positive. Thus, when the decision threshold is set too low, a model will commit false positive errors. However, in a near perfect classification model this will not happen until the curve has almost reached the 'perfect performance' point.

In Figure 3(c) we see that the model also behaves very nicely for a large range of decision threshold values. However, compared to the model in part (b) it starts to commit false positive errors much earlier and therefore the slope of the curve in part (c) is flatter. Another way of stating this is, that there exists no decision threshold for which the model is able to separate the classes perfectly. The model in Figure 4(d) is not only inferior because its curve is the farthest away from the 'perfect performance' point but we can observe that for large ranges of the ranking values the model commits more false positive errors than it provides true positive classifications. This shows up as concavities in the curve indicating that for certain ranges of the decision threshold the classification model performs worse than a random classifier.

Model Selection

A key notion in model assessment is model selection, that is, given two or more classification models, we need to pick one in order to be deployed. The criterion to pick one model over the other(s) has to answer two fundamental questions: (a) it needs to be general enough to describe model performance over a broad range of possible scenarios and (b) it needs to be able to discern whether the performance difference between models is statistically significant. It turns out that ROC curves answer both of these questions in a highly visual manner. Consider Figure 5(a), here we have two classifiers plotted in a ROC graph together with their respective 95% confidence bands (vertical bars) (Macskassy & Provost, 2004). It is easy to see that the curve that stretches almost into the top left corner represents the performance of the superior model (for a tpr = 0.9 this model commits virtually no false positives). In addition, because the confidence bands of the two curves are clearly separated we can state that the performance difference between the two models is statistically significant. In Figure 5(b) the situation is not so clear-cut. We again have two classifiers and the curve that reaches closer to the top left corner of the graph denotes the better performing model. However, since the confidence bands overlap, the performance difference between these two models is not statistically significant. In addition, closer inspection reveals that the confidence band for the upper curve is slightly wider than for the lower curve suggesting greater variability in the performance of the better performing model.

Figure 5. ROC curves with 95% confidence bands. (a) Two classifiers with a statistically significant difference in their performance. (b) Two classifiers whose difference in performance is not statistically significant



Future Trends

ROC analysis enjoys a continued growth of interest. Since 2004 there have been regularly scheduled workshops, the *Workshops on ROC Analysis in Machine Learning* (ROCML), which bring together an international group of researchers. Robust tools such as the ROCR package[3] for the R environment (Sing, Sander, Beerenwinkel, & Lengauer, 2005) contribute to the rapid adoption of ROC analysis as the preferred model analysis technique. At a technical level, the most important development is the extension of this analysis technique from binary classification problems to multi-class problems providing a much wider applicability of this technique (Everson & Fieldsend, 2006; Lane, 2000; Srinivasan, 1999).

CONCLUSION

Although brief, we hope that this overview provided an introduction to the fact that ROC analysis provides a powerful alternative to traditional model performance assessment using confusion matrices. We have shown that in contrast to traditional scalar performance metrics such as accuracy, recall, and precision derived from the confusion matrix, ROC analysis provides a highly visual account of a model's performance over a range of possible scenarios. We have also shown that ROC analysis is robust with respect to class skew, making it a reliable performance metric in many important application areas where highly skewed data sets are common (e.g. fraud detection).

REFERENCES

Bishop, C. (1995). *Neural networks for pattern recognition.*Oxford University Press, USA.

Breiman, L. (1996). *Bias, variance and arcing classifiers* No. Tech. Report 460, Statistics Dept., University of California.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees.*Wadsworth.

Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to support vector machines and other kernel-based learning methods.*Cambridge University Press.

Egan, J. P. (1975). *Signal detection theory and ROC analysis.*Academic Press New York.

Everson, R. M., & Fieldsend, J. E. (2006). Multi-class ROC analysis from a multi-objective optimisation perspective. *Pattern Recognition Letters, Special Number on ROC Analysis in Pattern Recognition,*

Fawcett, T. (2004). ROC graphs: Notes and practical considerations for researchers. *Machine Learning, 31*

Fawcett, T., & Flach, P. A. (2005). A response to webb and Ting's on the application of ROC analysis to predict classification performance under varying class distributions. *Machine Learning, 58*(1), 33-38.

Flach, P. (2004). *Tutorial at ICML 2004: The many faces of ROC analysis in machine learning.* Unpublished manuscript.

Flach, P., Blockeel, H., Ferri, C., Hernandez-Orallo, J., & Struyf, J. (2003). Decision support for data mining: Introduction to ROC analysis and its applications. *Data mining and decision support: Aspects of integration and collaboration* (pp. 81-90)

Flach, P. A. (2003). The geometry of ROC space: Understanding machine learning metrics through ROC isometrics. *Proceedings of the Twentieth International Conference on Machine Learning,* 194–201.

Hastie, T., Tibshirani, R., & Friedman, J. H. (2001). *The elements of statistical learning: Data mining, inference, and prediction.*Springer.

Lane, T. (2000). Extensions of ROC analysis to multi-class domains. *ICML-2000 Workshop on Cost-Sensitive Learning,*

Macskassy, S., & Provost, F. (2004). Confidence bands for ROC curves: Methods and an empirical study. *Proceedings of the First Workshop on ROC Analysis in AI (ROCAI-2004) at ECAI-2004,* Spain.

Mitchell, T. M. (1997). *Machine learning*McGraw-Hill Higher Education.

Provost, F., Fawcett, T., & Kohavi, R. (1998). The case against accuracy estimation for comparing induction

algorithms. *Proceedings of the Fifteenth International Conference on Machine Learning*, , 445–453.

Sing, T., Sander, O., Beerenwinkel, N., & Lengauer, T. (2005). ROCR: Visualizing classifier performance in R. *Bioinformatics (Oxford, England), 21*(20), 3940-3941.

Srinivasan, A. (1999). *Note on the location of optimal classifiers in n-dimensional ROC space* (Technical Report No. PRG-TR-2-99). Oxford, England: Oxford University Computing Laboratory.

Swets, J. A., Dawes, R. M., & Monahan, J. (2000). Better decisions through science. *Scientific American; Scientific American, 283*(4), 82-87.

KEY TERMS

Class Skew: In probability theory and statistics skewness is a measure of asymmetry of a distribution. Class skew refers to the asymmetry of the class distribution.

Classification Model/Classifier: A mathematical construct such as a decision tree or neural network that models the relationship between independent and dependent variables of a classification problem. Once such a model has been constructed it can be used to predict classifications on new data instances.

Confusion Matrix: A table that relates actual and predicted classifications by a model.

Model Assessment/Evaluation: The process of evaluating the key performance characteristics of a classification model. This is usually done within the context of a problem domain with specific model performance requirements.

Model Selection: The process of selecting a model from a set of potential models given a specific classification problem. The selection is usually based on a specific set of performance metrics dictated by the problem domain.

Optimal Operating Point: The point on the ROC curve where a model has the largest true positive rate while committing the smallest number of false positives.

Performance Metric: A performance-related measurement.

Receiver Operating Characteristic (ROC) Curve: A cost-benefit plot that describes the performance of a classification model.

ENDNOTES

1 Figures (b), (c), and (d) due to Peter Flach, *ICML'04 tutorial on ROC analysis*, International Conference on Machine Learning, 2004 (P. Flach, 2004).

2 Based on the algorithm published by Tom Fawcett (Fawcett, 2004).

3 We used the ROCR package for this work.

Modeling Quantiles

Claudia Perlich
IBM T.J. Watson Research, USA

Saharon Rosset
IBM T.J. Watson Research, USA

Bianca Zadrozny
Universidade Federal Fluminense, Brazil

INTRODUCTION

One standard Data Mining setting is defines by a set of n observations on a variable of interest Y and a set of p explanatory variables, or features, $x = (x_1,...,x_p)$, with the objective of finding a 'dependence' of Y on x. Such dependencies can either be of direct interest by themselves or used in the future to predict a Y given an observed x. This typically leads to a model for a conditional central tendency of $Y|x$, usually the mean $E(Y|x)$. For example, under appropriate model assumptions, Data Mining based on a least squares loss function (like linear least squares or most regression tree approaches), is as a maximum likelihood approach to estimating the conditional mean.

This chapter considers situations when the value of interest is not the conditional mean of a continuous variable, but rather a different property of the conditional distribution $P(Y|x)$, in particular a specific quantile of this distribution. Consider for instance the 0.9^{th} quantile of $P(Y|x)$, which is the function $c(x)$ such that $P(Y<c(x)|x) = 0.9$. As discussed in the main section, these problems (of estimating conditional mean vs. conditional high quantile) may be equivalent under simplistic assumptions about our models, but in practice they are usually not. We are typically interested in modeling extreme quantiles because they represent a desired 'prediction' in many business and scientific domains. Consider for example the motivating Data Mining task of estimating customer wallets from existing customer transaction data, which is of great practical interest for marketing and sales. A customer's wallet for a specific product category is the total amount this customer can spend in this product category. The vendor observes what the customers actually bought from him in the past, but does not typically have access to the customer's budget allocation decisions, their spending with competitors, etc. Information about customer's wallet, as an indicator of their potential for growth, is considered extremely valuable for marketing, resource planning and other tasks. For a detailed survey of the motivation, problem definition, see Rosset et al. 2005. In that paper we propose the definition of a customer's REALISTIC wallet as the 0.9^{th} or 0.95^{th} quantile of their conditional spending - this can be interpreted as the quantity that they may spend in the best case scenario. This task of modeling what a vendor can hope for rather than could expect turns out to be of great interest in multiple other business domains, including:

- When modeling sales prices of houses, cars or any other product, the seller may be very interested in the price they may aspire to get for their asset if they are successful in negotiations. This is clearly different from the 'average' price for this asset and is more in line with a high quantile of the price distribution of equivalent assets. Similarly, the buyer may be interested in the symmetric problem of modeling a low quantile.
- In outlier and fraud detection applications we may often have a specific variable (such as total amount spent on a credit card) whose degree of 'outlyingness' we want to examine for each one of a set of customers or observations. This degree can often be well approximated by the quantile of the conditional spending distribution given the customer's attributes. For identifying outliers we may just want to compare the actual spending to an appropriate high quantile, say 0.95.
- The opposite problem of the same notion of 'how bad can it get' is a very relevant component of financial modeling and in particular Value-at-Risk (Chernozhukov and Umantsev, 2001).

Addressing this task of quantile predictions, various researches have proposed methods that are often adaptations of standard expected value modeling approaches to the quantile modeling problem, and demonstrated that their predictions are meaningfully different from traditional expected value models.

BACKGROUND

Building and Evaluating Quantile Models

This section reviews some of the fundamental statistical and algorithmic concepts underlying the two main phases of predictive modeling - model building and model evaluation and selection - when the ultimate data mining goal is to predict high quantiles. Let us start from the easier question of model evaluation and model selection: given several models for predicting high quantiles and an evaluation data set not used for modeling, how can we estimate their performance and choose among them? The key to this problem is finding a loss function which describes well our success in predicting high quantile and evaluate the performance using this loss function. Clearly, the most important requirement from a loss function for evaluation is that the model which always predicts the conditional quantile correctly will have the best expected performance. Such a loss function indeed exists (Koenker, 2005).

Define the quantile loss function for the p^{th} quantile to be:

$$L_p(y,\hat{y}) = \begin{cases} p(y-\hat{y}) & \text{if } y \geq \hat{y} \\ (1-p)(\hat{y}-y) & \text{otherwise} \end{cases} \quad (1)$$

Figure 1 shows the quantile loss function for $p \in \{0.2, 0.5, 0.8\}$. With p=0.5 this is just absolute error loss. Expected quantile loss is minimized by correctly predicting the (conditional) p^{th} quantile of the conditional distribution. That is, if we fix a prediction point x, and define $c_p(x)$ to be the p^{th} quantile of the conditional distribution of Y given x:

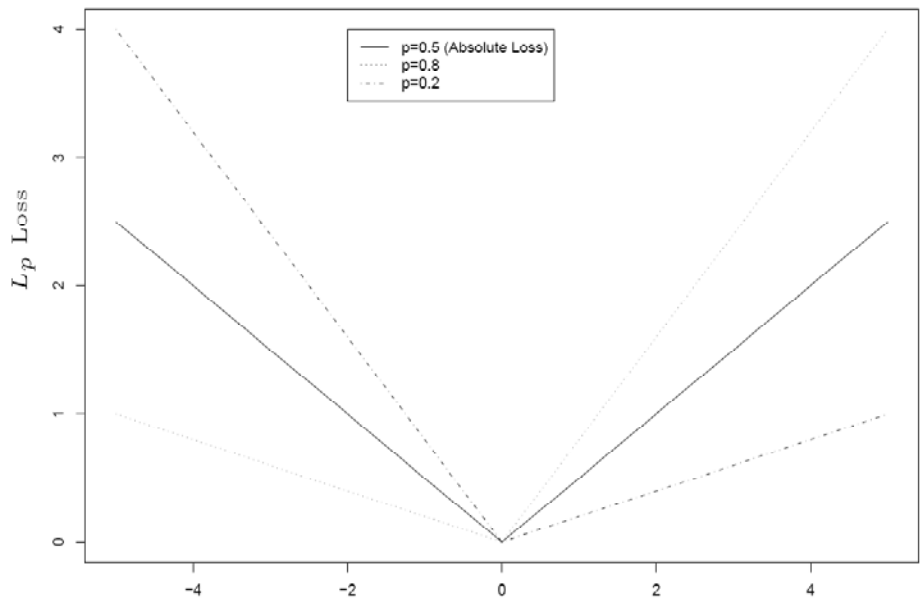
$$P(Y \leq c_p(x) \mid x) = p, \forall x$$

then the loss is optimized in expectation at every point by correctly predicting $c_p(x)$:

$$\arg\min_c E(L_p(Y,c) \mid x) = c_p(x)$$

With p=0.5, the expected absolute loss is minimized by predicting the median, while when p=0.9 we are in

Figure 1. Quantile loss functions for some quantiles



fact evaluating a model's ability to correctly predict the 90th percentile of the distribution $P(Y|x)$.

A different approach to evaluation is to look at the proportion of positive and negative residuals on the holdout data. A perfect prediction model for the 0.9th quantile will predict a value that is higher than the actual observed holdout response 90% of the time, on average. Thus we can examine whether the actual percentage of the time that the predictions are higher than observed response is indeed close to that, as a way of 'evaluating' high-quantile models. This is dangerous, of course, because a model which predicts $+\infty$ 90% of the time and $-\infty$ the other 10% would be perfect according to this measure.

On the modeling side, our first observation is that any property of the conditional distribution $P(Y|x)$ can be estimated well if we estimate well the whole distribution. In particular, if we have a parametric model for $P(Y|x)$ which we believe is true and which we have enough data to estimate, then it is often the best policy to apply all our effort towards estimating this model's parameters well (e.g., using a maximum likelihood approach), regardless of what property of $P(Y|x)$ we are ultimately interested in. For example, if we believe that $P(Y|x)$ is homoscedastic Gaussian and $E(Y|x) = \beta x$ is linear in x, then a maximum likelihood approach would call for fitting a linear regression model of Y on x. Furthermore, this would also trivially imply that the 0.9th quantile of $P(Y|x)$ is linear in x, and is simply $E(Y|x)$+ constant.

However, parametric assumptions are usually over-simplifications of realistic modeling problems, especially those encountered in complex data-mining domains, and one should either dispose with them completely (and choose non-parametric approaches), or treat them with skepticism. An alternative to the parametric approach is to build the model by minimizing an 'empirical risk' over the training data, which represents well the prediction task. In the case of quantile modeling, the quantile estimation loss function certainly qualifies (a similar approach leads Friedman et al. 2000 to advocate the logistic regression loss function for boosting, for example).

In practice, both of these approaches may have advantages and disadvantages. An additional consideration is one of variance, especially when modeling high quantiles - does the high-quantile loss function allow us to make efficient use of the data for modeling?

MAIN FOCUS

Over the recent past, the modeling of quantiles has received increasing attention. The modeling objectives were either prediction or to gain insights how the statistical dependencies for quantiles differ from expected value models. We are aware several such quantile estimation methods, of which we discuss two below. In addition, we present how two of the best studied and also practically most common modeling approaches in machine learning (k-nearest neighbors and regression trees) can be adjusted to model the quantiles.

Linear Quantile Regression

A standard technique for quantile regression that has been developed and extensively applied in the Econometrics community is linear quantile regression (Koenker, 2005). In linear quantile regression, we assume that the conditional quantile function is a linear function of the explanatory variables of the form $\beta \cdot x$ and we estimate the parameters $\hat{\beta}$ that minimize the quantile loss function (Equation 1). It can be shown that this minimization is a linear programming problem and that it can be efficiently solved using interior point techniques (Koenker, 2005). Implementations of linear quantile regression are available in standard statistical analysis packages such as R and SAS. The obvious limitation of linear quantile regression is that the assumption of a linear relationship between the explanatory variables and the conditional quantile function may not be true. To circumvent this problem, Koenker (2005) suggests using nonlinear spline models of the explanatory variables. Recently, a kernel quantile regression approach has also been proposed for the same purpose.

Kernel Quantile Estimation

Takeuchi et al. (2005) have recently proposed a technique for nonparametric quantile estimation that applies the two standard features of kernel methods to conditional quantile estimation: regularization and the kernel trick. They show that a regularized version of the quantile loss function can be directly minimized using standard quadratic programming techniques. By choosing an appropriate kernel, such as a radial basis function kernel, one can obtain nonlinear conditional quantile estimates. They compare their method experimentally

to linear quantile regression and to the nonlinear spline approach suggested by Koenker (2005) on many small datasets for different quantiles and find that it performs the best in most cases.

The main disadvantage of this technique as a data mining method is that it is not scalable, as the computational time grows super-linearly with the number of examples. This is worsened by the fact that the kernel parameters have to be chosen through cross-validation on the training data. Such limitations render the approach unsuitable for most realistic modeling problems.

Quantile k-Nearest Neighbor

The traditional k-nearest neighbor model (kNN) is defined as

$$\hat{y}(x) = 1/k \sum_{x_i \in N_k(x)} y \tag{2}$$

where $N_k(x)$ is the neighborhood of x defined by the k closes points x_i in the training sample for a given distance measure (e.g., Eucledian). From a statistical perspective we can view the set $y_j \in N_k(x)$ as a sample from the approximated conditional distribution of $P(Y|x)$. The standard kNN estimator of $\hat{y}$ is simply the expected value of this conditional distribution approximated by a local neighborhood. For quantile estimation we are not interested in the expected value (i.e., an estimate of $E(Y|x)$) but rather a particular quantile $c_p(x)$ of the conditional distribution $P(Y|x)$ such that $P(Y>c_p(x)|x)=q$. Accordingly we can estimate $c_p(x)$ in a k-nearest neighbor setting as the q^{th} quantile of the empirical distribution of $\{y_j : x_j \in N_k(x)\}$. If we denote that empirical distribution by:

$$\hat{G}_x(c) = 1/k \sum_{x_j \in N_k(x)} 1\{y_j \le c\} \tag{3}$$

then our kNN estimate of the q^{th} quantile of $P(Y|x)$ would be $\hat{G}_x^{-1}(q)$.

The interpretation is similarly that the values of Y in the neighborhood $N_k(x)$ are a sample from the conditional distribution $P(Y|x)$ and we are empirically estimating its q^{th} quantile. An important practical aspect of this estimate is that, in contrast to the standard kNN estimates, it imposes a constraint on k. While k=1 produces an unbiased (while high variance) estimate of the expected value, the choice of k has to be at least 1/(1-q) to provide an upper bound for the estimate of the q^{th} 'high' quantile (more generally we have $k \ge \max(1/q, 1/(1-q))$).

The definition of neighborhood is determined based on the set of variables, the distance function and implicit properties such as scaling of the variables. The performance of a kNN model is very much subject to the suitability of the neighborhood definition to provide a good approximation of the true conditional distribution - this is true for the standard problem of estimating the conditional mean and no less so for estimating conditional quantiles.

Quantile Regression Tree

Tree-induction algorithms are very popular in predictive modeling and are known for their simplicity and efficiency when dealing with domains with large number of variables and cases. Regression trees are obtained using a fast divide and conquer greedy algorithm that recursively partitions the training data into subsets. Therefore, the definition of the neighborhood that is used to approximate the conditional distribution is not predetermined as in the case of the kNN model but optimized locally by the choice of the subsets. Work on tree-based regression models traces back to Morgan and Sonquist (1963) but the major reference is the book on classification and regression trees (CART) by Breiman et al. (1984). We will limit our discussion to this particular algorithm. Additional regression tree implementation include RETIS (Karalic, 1992), CORE (Obnik-Sikonja, 1997,) M5 (Quinlan, 1993), RT (Torgo, 1997).

A tree-based modeling approach is determined predominantly by three components:

1. The **splitting criterion** which is used to select the next split in the recursive partitioning,
2. The **pruning method** that shrinks the overly large tree to an optimal size after the partitioning has finished in order to reduce variance,
3. The **estimation method** that determines the prediction within a given leaf.

The most common choice for the splitting criterion is the least squares error (LSE). While this criterion is consistent with the objective of finding the conditional expectation, it can also be interpreted as a measure

of the improvement of the approximation quality of the conditional distribution estimate. Tree induction searches for local neighborhood definitions that provide good approximations for the true conditional distribution *P(Y|x)*. So an alternative interpretation of the LSE splitting criterion is to understand it as a measure of dependency between *Y* and an x_i variable by evaluating the decrease of uncertainty (as measured by variance) through conditioning. In addition, the use of LSE leads to implementations with high computational efficiency based on incremental estimates of the errors for all possible splits.

Pruning is the most common strategy to avoid overfitting within tree-based models. The objective is to obtain a smaller sub-tree of the initial overly large tree, excluding those lower level branches that are unreliable. CART uses Error-Complexity pruning approach which finds a optimal sequence of pruned trees by sequentially eliminating the subtree (i.e., node and all its ancestors) that minimizes the increase in error weighted by the number of leaves in the eliminated subtree:

$$g(t,T_t) = \frac{Err(t) - Err(T_t)}{S(T_t) - 1}$$

where $Err(T_t)$ is the error of the subtree T_t containing t and all its ancestors, and *Err(t)* is the error if it was replaced by a single leaf, and $S(T_t)$ is the number of leaves in the subtree. *Err()* is measured in terms of the splitting criterion (i.e., for standard CART it is squared error loss). Given an optimal pruning sequence, one still needs to determine the optimal level of pruning and Breiman et al. (1984) suggest cross validation on a holdout set.

Finally CART estimates the prediction for a new case that falls into leaf node *l* similarly to the kNN algorithm as the mean over the set of training responses D_l in the leaf:

$$\hat{y}_l(x) = \frac{1}{n_l} \sum_{y_j \in D_l} y_i$$

where n_l is the cardinality of the set D_l of training cases in the leaf. Given our objective of quantile estimation, the most obvious adjustment to CART is to replace the sample mean estimate in the leaves with the quantile estimate using the empirical local estimate $\hat{G}_{D_l}(c)$ of *P(Y|x)* as in equation (3).

In initial empirical work we observe that the empirical results suggest that changing the splitting to quantile loss does not improve the predictive performance of quantile trees. This is consistent with the interpretation of LSE as a measure of variance-reduction and thereby of the dependency between *Y* and an x_i.

FUTURE TRENDS

The importance of quantile modeling in Data Mining application areas will continue to grow as more and more researchers and practitioners discover the utility of moving beyond conditional-mean estimation, into estimation of conditional quantiles, as a way of expressing optimistic aspirations (like in the wallet estimation problem) or conservative loss estimates (like when modeling value at risk). Some of the interesting research questions that need to be addressed include:

- Can effective quantile modeling approaches be devised for extreme quantiles (like 0.999), to serve in outlier detection, fraud analysis etc.
- What is the effect of the strong hetero-skedasticity often present in the data we model in Data Mining applications on the effectiveness of conditional quantile estimation? How does it compare to the effect on conditional mean estimation? Can we describe the conditions under which quantile modeling may be a better approach in general?

CONCLUSION

In this chapter we presented a new class of quantile modeling problems that are highly relevant in many business domains, including wallet estimation, price/salary prediction and value-at-risk. We reviewed the statistical considerations involved in designing methods for high-quantile estimation and described some existing quantile modeling methods, as well as our own adaptations of kNN and CART to quantile modeling.

In summary, there is a set of algorithms readily available to address the relevant issue of quantile modeling. The necessary adjustments to classical machine learning techniques such as tree induction are straight forward and result in reliable, interpretable, and efficient solutions.

REFERENCES

Breiman, L., (1996). Bagging predictors. *Machine Learning*, 24(2), 123-140.

Breiman, L., J.H. Friedman, Olshen, R.A., & C.J. Stone. (1984) *Classification and regression trees*. Wadsworth International Group.

Chernozhukov, V., & Umantsev, L. (2001). Conditional Value-at-Risk: Aspects of Modeling and Estimation, *Empirical Economics*, 26 (1), 271-292.

Friedman, J., T.Hastie, & R.Tibshirani, (2000). Additive logistic regression: A statistical view of boosting. *Annals of Statistics*, 337-374.

Hammond, J.S., Keeney, R.L., & Raiffa, H. (1998). The hidden traps in decision making. *Harvard Business Review*, 76(5).

Kahneman D., & A.Tversky. (1973). On the psychology of prediction. *Psychology Review*, 80, 237-251.

Karalic A., (1992). Employing linear regression in regression tree leaves. *In Proceedings of the European Conference on Artificial Intelligence*, 440-441. John Wiley & Sons.

Koenker R., (2005). *Quantile Regression. Econometric Society Monograph Series*. Cambridge University Press.

Langford, J., R.Oliveira, & B.Zadrozny (2006). Predicting the median and other order statistics via reduction to classification. *Submitted to UAI-2006.*

Morgan & Sonquist (1963). Problems in the analysis of survey data and a proposal. *JASA*, 58, 415-434.

Quinlan, R., (1993) Combining instance-based and model-based learning. *In Proceedings of the Tenth International Conference of Machine Learning*, 236-243. Morgan Kaufmann.

Robnik-Sikonja, M., (1997). CORE - a system that predicts continuous variables. *In Proceedings of ERK.*

Rosset, S., C. Perlich, B. Zadrozny, S. Merugu, S. Weiss, & R. Lawrence. (2005). Wallet estimation models. *In International Workshop on Customer Relationship Management: Data Mining Meets Marketing*.

Takeuchi, I., Le, Q.V., Sears, T., & Smola, A. (2005). *Nonparametric quantile regression*. NICTA Technical Report.

Torgo, L. (1997). Functional models for regression tree leaves. *In Proceedings of the 14th International Conference on Machine Learning*, 385-393. Morgan Kaufmann.

Wansink, B., Kent R.J., & Hoch S.J. (1998). An anchoring and adjustment model of purchase quantity decisions. *Journal of Marketing Research*, 35(1):71-81.

KEY TERMS

Customer Wallet Estimation: A modeling problem, of estimating how much money a company's customers could potentially be spending with it (as opposed to how much they are actually spending). It is important for marketing and planning purposes.

Linear Quantile Regression: Linear regression model of the form $\beta \cdot x$ where β is estimated subject to minimization of quantile loss using linear programming.

Nearest Neighbors and Regression Trees: Two of the most commonly used approaches for regression modeling. Both seek to define for every point $\boldsymbol{x}$ a 'neighborhood' of similar points and then predict the response Y at point $\boldsymbol{x}$ as the average of the neighborhood.

Quantile: Given a random variable Y, the p^{th} quantile of Y with $0 \leq p \leq 1$ is the value c such that $P(Y \leq c) = p$.

Quanting: Reduction approach to quantile estimation that constructs an ensemble of classifiers from observations that are weighted according to the quantile loss.

Quantile Loss: An asymmetric loss function, parameterized by p whose expected loss is minimized by correctly predicting the conditional p^{th} quantile. Thus, it is useful both for building and evaluating quantile models.

Quantile Modeling: A predictive modeling problem which seeks to model the conditional p^{th} quantile of the response Y given the features $\boldsymbol{x}$.

Quantile Nearest Neighbors and Quantile Regression Trees: Versions of these two methods that have been adapted to predicting quantiles of Y given $\boldsymbol{x}$, by changing either the neighborhood definition, or the averaging method, or both.

Modeling Score Distributions

Anca Doloc-Mihu
University of Louisiana at Lafayette, USA

INTRODUCTION

The goal of a web-based retrieval system is to find data items that meet a user's request as fast and accurately as possible. Such a search engine finds items relevant to the user's query by scoring and ranking each item in the database. Swets (1963) proposed to model the distributions of these scores to find an optimal threshold for separating relevant from non-relevant items. Since then, researchers suggested several different score distribution models, which offer elegant solutions to improve the effectiveness and efficiency of different components of search systems.

Recent studies show that the method of modeling score distribution is beneficial to various applications, such as outlier detection algorithms (Gao & Tan, 2006), search engines (Manmatha, Feng, & Rath, 2001), information filtering (Zhang & Callan, 2001), distributed information retrieval (Baumgarten, 1999), video retrieval (Wilkins, Ferguson, & Smeaton, 2006), kernel type selection for image retrieval (Doloc-Mihu & Raghavan, 2006), and biometry (Ulery, Fellner, Hallinan, Hicklin, & Watson, 2006).

The advantage of the score distribution method is that it uses the statistical properties of the scores, and not their values, and therefore, the obtained estimation may generalize better to not seen items than an estimation obtained by using the score values (Arampatzis, Beney, Koster, & van der Weide, 2000). In this chapter, we present the score distribution modeling approach, and then, we briefly survey theoretical and empirical studies on the distribution models, followed by several of its applications.

BACKGROUND

The primary goal of information retrieval is to retrieve all the documents which are relevant to a user query, while retrieving as few non-relevant documents as possible (Baeza-Yates & Ribeiro-Neto, 1999). This is achieved by ranking the list of documents according to their relevance to the user's query. Since relevance is a subjective attribute, depending on the user's perception of the closeness between the user submitted query and the real query from her or his mind, building a better way to retrieve data is a challenge that needs to be addressed in a retrieval system.

In other words, a retrieval system aims at building the request (query) that best represents the user's information need. This optimal request is defined by using an explicit data-request matching (Rocchio, 1971) that should produce a ranking in which all relevant data are ranked higher than the non-relevant data. For the matching process, a retrieval system uses a retrieval function, which associates each data-query pair with a real number or score (the retrieval status value). Then, the retrieval system uses these scores to rank the list of data.

However, researchers (Swets, 1963; Arampatzis, Beney, Koster, & van der Weide, 2000; Manmatha, Feng, & Rath, 2001) raised the question of whether or not the statistical properties of these scores, displayed by the shape of their distribution, for a given query, can be used to model the data space or the retrieval process. As a result, they proposed and empirically investigated several models of the score distributions as solutions to improve the effectiveness and efficiency of the retrieval systems. The next section introduces the score distribution method.

MAIN FOCUS

The Score Distribution Method

The probability ranking principle (Robertson, 1977) states that a search system should rank output in order of probability of relevance. That is, the higher the score value of the document, the more relevant to the query is considered the document to be. In the binary relevance case, which is the case we are interested in, the ideal retrieval system associates scores to the relevant and non-relevant data such that the two groups are well

separated, and relevant data have higher scores than the non-relevant data. In practice, retrieval systems are not capable to completely separate the relevant from the non-relevant data, and therefore, there are non-relevant data with higher score values than those of some relevant data.

The score distribution method tries to find a good way to separate these two groups of data by using statistical properties of their scores. The method assumes that the relevant and non-relevant data form two separate groups, with each group being characterized by its own characteristics different from the other group. For each group, the method plots the corresponding score values within the group, and then, tries to find the shape of the curve generated by these scores. In fact, this curve is approximated with a distribution usually chosen via experimental results (the best fit from a set of known distributions, such as normal, exponential, Poisson, gamma, beta, Pareto). Once the two distributions are known (or modeled), they are used to improve the search system.

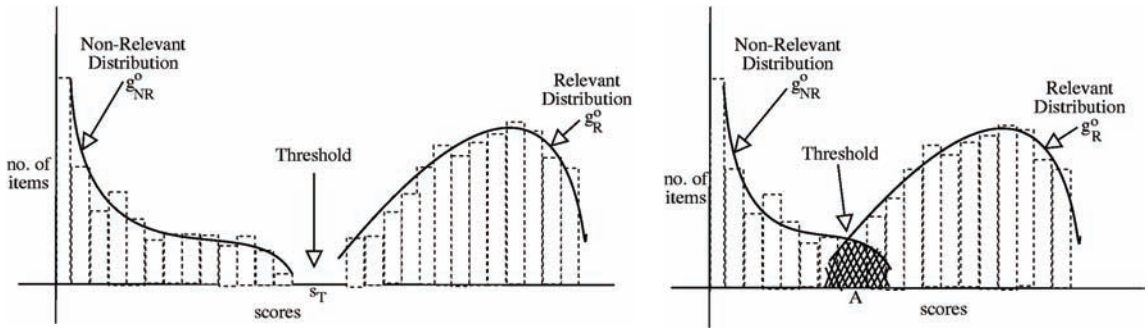
Figure 1 illustrates the score distribution method, (a) in the ideal case, when the relevant and non-relevant data are well separated by the retrieval system, and (b) in a real case, when there are non-relevant data with score values higher than those of some relevant data. The scores of non-relevant data are grouped toward the left side of the plot, and the scores of relevant data are grouped toward the right side of the plot. A curve shows the shape of the score distribution of each group (of relevant and non-relevant data, respectively). Note that, in this figure, the two curves (given as densities $g^{O}_{R}(s)$ and $g^{O}_{NR}(s)$) do not display any particular distribution; they represent the curves of some arbitrary distributions. Basically, the score distribution method consists in choosing the best possible shapes of the distributions of the two groups. Then, any relevant (non-relevant) data is assumed to follow its chosen relevant (non-relevant) distribution.

Ideally, the two distribution curves do not meet (Figure 1 (a)), but in reality, the two curves meet at some point. However, as shown in Figure 1 (b), there is a common region between the two score distribution curves (named A). This area is of most interest for researchers; it includes relevant data with score values very close (lower or not) to the score values of non-relevant data. Therefore, by finding a way to minimize it, one finds a way to approximately separate the two data. Another solution is to find a threshold that separates optimally the relevant data from non-relevant ones.

The advantage of the score distribution method is that it uses the statistical properties of the scores (the shape of their distribution) and not their values, which conducts to an estimation of the threshold or the area A (Figure 1 (b)) that may generalize better to not seen data than an estimation method, which uses the score values (Arampatzis, Beney, Koster, & van der Weide, 2000).

We presented the method in the case that the entire data from collection is used. However, for efficiency reason, in practice, researchers prefer to return to user only the top most relevant N data. In this case, as Zhang and Callan (2001) noticed, the method is

Figure 1. Score distributions for relevant and non-relevant data



(a) Ideal case, at which a retrieval system aims, with a clear separation between the relevant and non-relevant data.

(b) Real case, which shows a common region for scores of the relevant and non-relevant data.

biased, especially for low scoring items, which do not occur between these top N chosen items. However, for high scoring data, the model offers a relatively good estimation (Zhang & Callan, 2001; Manmatha, Feng, & Rath, 2001).

Models of Score Distributions

Since introduced by Swets (1963), researchers used various combinations of the two distributions. Some chose the same type of distribution for both groups of data, whereas others argued that these should have different shapes. For example, Swets (1963) proposed two normal distributions of equal variance and later, two unequal variance normals or two exponentials (Swets, 1969); Bookstein (1977) used two Poisson distributions; Baumgarten (1999) used two gamma distributions. From the proposed models that use different distributions, the Gaussian-exponential model, which uses a normal for relevant data and an exponential for non-relevant data, is the most used model (Arampatzis & van Hameren, 2001; Manmatha, Feng & Rath, 2001; Zhang, & Callan, 2001; Collins-Thompson, Ogilvie, Zhang & Callan, 2003; Gao & Tan, 2006; Wilkins, Ferguson & Smeaton, 2006; Doloc-Mihu & Raghavan, 2006).

As shown by these examples, researchers investigated different specific distributions, such as normal, exponential, gamma, Poisson, but, to date, there is no agreement on either one of them as being the best distribution that models the scores or either relevant or not-relevant data. As Robertson (2007) noted recently, "clearly a strong argument for choosing any particular combination of distributions is that it gives a good fit to some set of empirical data" (p. 40). However, researchers addressed this issue in two ways. Some researchers base their models on the empirical evidence, while others try to find theoretical evidence. In the following, we briefly present such recent work on score distributions.

Theoretical Advances on Score Distribution Models

Rijsbergen (1979) observed that for search engines like SMART there is no evidence that the two score distributions should have similar shapes or that they follow Gaussian distributions as proposed by Swets (1963). Recently, some researchers try to find theoretical evidence to this observation. For example, Madigan, Vardi & Weissman (2006) presented an interesting mathematical analysis on the different combinations of distributions. They applied the extreme value theory (Resnick, 1987) to study why early precision increases as collection size grows. Their analysis showed that the asymptotic behavior of two retrieval measures (of effectiveness), P@K, the proportion of the top K documents that are relevant, and C@K, the number of non-relevant documents amongst the top K relevant documents, depends on the score distributions and on the relative proportion between relevant and non-relevant documents in the collection. The results contradict the normal-exponential model of Manmatha et al. (2001), and sustain Swets (1963) model with the remark that different choices (like, exponential-exponential, Pareto-Pareto, Beta-Beta) can result in early precision approaching zero or one or a constant as the number of ranked documents increases.

Robertson (2007) proposes the convexity hypothesis, which is a generalization of the hypothesis of the inverse recall-precision relationship and states that "for all good systems, the recall-fallout curve is convex" (p. 43). This hypothesis can be formulated as a condition on the probability of relevance of a document at an exact score: the higher the score, the higher the probability of relevance. The author proves that models like exponential-exponential (Swets, 1969), normal-normal with equal variances (Swets, 1963), Poisson-Poisson (Bookstein, 1977), gamma-gamma (Baumgarten, 1999) for certain settings of the parameters b and c, hold the convexity condition, and that the normal-normal model with different variances and the normal-exponential model (Manmatha, Feng & Rath, 2001) violate the condition. In conclusion, this theoretical result shows that the distribution models, which do not hold the convexity hypothesis, do not provide general solutions, but they are just reasonable approximations to the real distributions.

Empirical Studies on Score Distribution Models

Manmatha, Feng, & Rath (2001) show empirically that Gaussian-exponential models can fit approximately well the score distributions of the relevant and non-relevant documents corresponding to a given query. Moreover, they propose a model in which these score distributions are used to calculate the posterior prob-

abilities of relevance given the score via Bayes's rule. Experimental results on TREC-3, TREC-4, and TREC-6 data show that this method works for both probabilistic search engines, like INQUERY, and vector space search engines, like SMART, but it offers only an empirical approximation to the real distributions. In the same study, the authors also show that when relevance information is not available, these distributions can be recovered via the expectation-maximization algorithm by fitting a mixture model consisting of a Gaussian and an exponential function.

Applications of Score Distribution Method

As noted by Manmatha et al. (2001), once known, the score distributions can be used to map the scores of a search engine to probabilities. Score distributions can be beneficial to several tasks. A first example concerns the combination of the outputs of different search engines operating on one or more databases in different languages or not. This combination can be performed for example, by averaging the probabilities, or by using the probabilities to select the best engine for each query (Manmatha, Feng, & Rath, 2001).

Another example deals with the task of filtering thresholds. Here, Arampatzis and van Hameren (2001) proposed a score-distributional threshold optimization method for adaptive binary classification tasks. The method was tested on the TREC-9 Filtering Track and obtained the best results when using a Gaussian to model the distribution scores of the relevant documents, and an exponential for the distribution scores of the non-relevant documents. Zhang and Callan (2001) propose an algorithm that addresses the bias aspect of training data in information filtering, which happens because relevant information is not available for documents with scores below the threshold. Based on the Maximum Likelihood Principle, this algorithm estimates the parameters of the two score distributions (Gaussian-exponential model) and the ratios of the relevant and the non-relevant documents. The authors report significant improvement on the TREC-9 Filtering Track.

Baumgarten (1999) proposed a probabilistic solution based on a gamma-gamma model to select and fuse information from document subcollections over a distributed document collection. The model integrates acceptable non-heuristic solutions to the selection and fusion issues in a distributed environment, and shows encouraging experimental results that outperforms its non-distributed counterpart.

Gao and Tan (2006) proposed two approaches to convert output scores from outlier detection algorithms into well-calibrated probability estimates. Their second approach is similar to the one proposed by Manmatha et al. (2001) for search engines; it models the score distributions of outliers as a mixture of a Gaussian and an exponential probability function and calculates the posterior probabilities via Bayes's rule. The reported results show that the method helps in improving the selection of a more appropriate outlier threshold, and in improving the effectiveness of an outlier detection ensemble. Also, as in the case of search engines, the missing labels of outliers can be considered as hidden variables that can be learnt via the expectation-maximization algorithm together with the distribution model parameters.

Recently, the score distribution method was applied to multimedia data. For example, Doloc-Mihu and Raghavan (2006) used score distribution models to address the problem of automatically selecting the kernel type (Cristianini & Shawe-Taylor, 2000; Chapelle, Haffner, & Vapnik, 1999) for a given query in image retrieval. The authors empirically observed a correlation between the different kernel types and the different shapes of the score distributions. The proposed method selects the kernel type for which the surface of the intersection area (A) between the two distributions (see Figure 1 (b)) is minimal. The best retrieval results were obtained for the Gaussian-exponential model of the relevant and non-relevant images represented by color histograms in RGB color space. Further, this model gave also the best fit for fused multi-modal data (Doloc-Mihu & Raghavan, 2007).

Wilkins, Ferguson and Smeaton (2006) proposed a model based on score distributions to automatically generate the weights needed for multi-modal data fusion in video retrieval. Their model was formulated based on empirical observations, and compares features according to the score distribution of the scores of data returned by them on a per query basis. Reported results on TRECVid 2004 and 2005 collections demonstrate the applicability of the model.

Finally, we mention the study performed recently by Ulery et al. (2006) of score-level fusion techniques involving different distribution models for biometric data. The authors used fingerprint and face data to evalu-

ate the effectiveness of eight score fusion techniques. The study concluded that fusing scores is effective, but it depends on a series of factors such as the ability to model score distributions accurately.

FUTURE TRENDS

Future research should focus on developing a well-founded theoretical model for choosing the score distributions that describes the correlations, empirically observed, between score distributions and relevance. Further, such a model should be tested on real-world data to investigate its entire potential.

Different Data Mining tasks, such as mining Web information, mining multimedia, and biomedical data, and information retrieval tasks, such as multi-lingual retrieval, and relevance feedback may benefit from modeling score distributions. Other potential applications such as feature selection and fusion for image, video and sound retrieval need to be considered for multimedia engines.

CONCLUSION

In this chapter, we presented the score distribution modeling approach, which offers elegant solutions to improve the effectiveness and efficiency of different components of search systems. We surveyed several such models used for various tasks, such as ensemble outlier detection, finding thresholds for document filtering, combining the outputs of different search engines, selecting the kernel type in an Image Retrieval System, and fusion of multimodal data in video retrieval and biometry. These applications demonstrate, mostly through empirical testing, the potential of the score distribution models to real world search systems.

REFERENCES

Arampatzis, A., Beney, J., Koster, C., & van der Weide, T.P. (2000). Incrementally, Half-Life, and Threshold Optimization for Adaptive Document Filtering. *IEEE Transactions on Knowledge and Data Engineering*, 15(6), 1460-1471.

Arampatzis, A., & van Hameren, A. (2001). The Score-Distributional Threshold Optimization for Adaptive Binary Classification Tasks. In W.B. Croft, D.J. Harper, D.H. Kraft, J. Zobel (Eds.), *Proceedings of the 24th ACM SIGIR Conference on Research and Development in Information Retrieval* (pp.285-293). New York: ACM Press.

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern Information Retrieval*, New York: Addison-Wesley.

Baumgarten, C. (1999). A Probabilistic Solution to the Selection and Fusion Problem in Distributed Information Retrieval. In M. Hearst, F. Gey, R. Tong (Eds.), *Proceedings of the 22nd ACM SIGIR Conference on Research and Development in Information Retrieval, Univ. of California at Berkeley, USA,* (pp.1-8). New York: ACM Press.

Bookstein, A. (1977). When the most 'pertinent' document should not be retrieved – an analysis of the Swets model. *Information Processing and Management*, 13(6), 377-383.

Chapelle, O., Haffner, P., & Vapnik, V. (1999). SVMs for Histogram-Based Image Classification. *IEEE Transactions on Neural Networks*, 10(5), 1055-1064.

Collins-Thompson, K., Ogilvie, P., Zhang, Y., & Callan, J. (2003). Information filtering, novelty detection and named page finding. In E.M. Voorhees, L.P. Buckland (Eds.), *The 11th Text retrieval Conference, TREC 2002, NIST Special Publication 500-251 (NIST 2003), Gaithersburg, Maryland, USA,* (pp. 107-118).

Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to Support Vector Machines and other kernel-based learning methods*, New York: Cambridge University Press.

Del Bimbo, A. (2001). *Visual Information Retrieval*, San Francisco, CA: Morgan Kaufmann Publishers, Inc.

Doloc-Mihu, A. (2007). *Adaptive Image Retrieval: Similarity Modeling, Learning, Fusion, and Visualization*, Ph.D. dissertation. Louisiana, USA: University of Louisiana at Lafayette.

Doloc-Mihu, A., & Raghavan, V. V. (2006). Score Distribution Approach to Automatic Kernel Selection for Image Retrieval Systems. In F. Esposito, Z.W. Ras, D. Malerba, G. Semeraro (Eds.), *Proceedings of the*

16th International Symposium on Methodologies for Intelligent Systems (ISMIS 2006) Bari, Italy: LNAI, Vol.4203, Foundations of Intelligent Systems (pp. 443-452). Berlin: Springer-Verlag.

Doloc-Mihu, A., & Raghavan, V. V. (2007). Fusion and Kernel Type Selection in Adaptive Image Retrieval. In B.V. Dasarathy (Ed.), *Multisensor, Multisource Information Fusion, SPIE Defense and Security Symposium, (DSS 2007), Vol.6571, Orlando, Florida, USA* (pp. 75-90). Washington: SPIE Press.

Gao, J., & Tan, P.-N. (2006). Converting Output Scores from Outlier Detection Algorithms into Probability Estimates. In B. Werner (Ed.), *Proceedings of the 6th International Conference on Data Mining (ICDM 2006), Hong Kong,* (pp. 212-221). Los Alamitos: IEEE Computer Society.

Han, J., & Kamber, M. (2006). *Data Mining: Concepts and Technique,*. San Francisco, CA: Morgan Kaufmann Publishers, Inc.

Madigan, D., Vardi, Y., & Weissman, I. (2006). Extreme Value Theory Applied to Document Retrieval from Large Collections. *Information Retrieval,* 9(3), 273-294.

Manmatha, R., Feng, F., & Rath, T. (2001). Using Models of Score Distributions in Information Retrieval. In W.B. Croft, D.J. Harper, D.H. Kraft, J. Zobel (Eds.), *Proceedings of the 24th ACM SIGIR Conference on Research and Development in Information Retrieval, New Orleans, USA,* (pp.267-275). New York: ACM Press.

Resnick, S.I. (1987). *Extreme Values, Regular Variation and Point Processes,* New York: Springer-Verlag.

Rijsbergen, C.J. (1979). *Information Retrieval,* Butterworths, London, 2nd Ed.. Retrieved December 9, 2007, from http://www.dcs.gla.ac.uk/Keith/Preface.html.

Robertson, S.E. (1977). The probability ranking principle in information retrieval. *Journal of Documentation,* 33(4), 294-304.

Robertson, S. (2007). On score distributions and relevance. In G. Amati, C. Carpineto, G. Romano (Eds.), *29th European Conference on Information Retrieval, ECIR 2007, Rome, Italy, LNCS, vol. 4425* (pp. 40-51). Berlin: Springer-Verlag.

Rocchio, J.J. (1971). Relevance feedback in Information Retrieval. In Gerard Salton (Ed.), *The SMART Retrieval System - Experiments in Automatic Document Processing* (pp. 313-323). Englewood Cliffs, New Jersey, USA: Prentice-Hall, Inc.

Swets, J. A. (1963). Information Retrieval Systems. *Science*, 141(3577), 245-250.

Swets, J. A. (1969). Effectiveness of Information Retrieval Methods. *American Documentation*, 20, 72-89.

Ulery, B., Fellner, W., Hallinan, P., Hicklin, A., & Watson, C. (2006). Studies of Biometric Fusion. *NISTIR, 7346*. Retrieved October 10, 2007, from http://www.itl.nist.gov/

Zhang, Y., & Callan, J. (2001). Maximum Likelihood Estimation for Filtering Thresholds. In W.B. Croft, D.J. Harper, D.H. Kraft, J. Zobel (Eds.), *Proceedings of the 24th International ACM SIGIR Conference on Research and Development in Information Retrieval, New Orleans, USA,* (pp.294-302). New York: ACM Press.

Wilkins, P., Ferguson, P., & Smeaton, A. F. (2006). Using Score Distributions for Query-time Fusion in Multimedia Retrieval. In J.Z. Wang, N. Boujemaa (Eds.), *Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval, MIR 2006, Santa Barbara, California, USA,* (pp. 51-60). New York: ACM Press.

Wikipedia (2007a). Information Retrieval. Retrieved December 8, 2007, from http://en.wikipedia.org/wiki/Information_retrieval.

Wikipedia (2007b). Kernel Methods. Retrieved December 8, 2007, from http://en.wikipedia.org/wiki/Kernel_Method.

KEY TERMS

Expectation-Maximization Algorithm: An iterative algorithm used for finding maximum likelihood estimates of parameters in probabilistic models, where the model depends on unobserved hidden variables.

Fusion: Process of combining two distinct things. **Data fusion** and **Information Fusion** used in Data Mining are generally defined as the set of techniques that combine/merge data/information from multiple sources.

Information Filtering: Process of monitoring information in order to present to the user information items the user is interested in.

Information Retrieval: Science of searching for information in documents, searching for documents themselves, searching for metadata which describe documents, or searching within databases, whether relational stand-alone databases or hypertextually-networked databases such as the World Wide Web (Wikipedia, 2007a). Similarly, Image Retrieval is the science of searching and retrieving images from a large database of digital images (del Bimbo, 2001; Doloc-Mihu, 2007).

Kernel Methods: Class of algorithms for pattern analysis, which use kernel functions that allow to operate in the feature space without ever computing the coordinates of the data in that space, but rather by simply computing the inner products between the corresponding features of all pairs of data in the feature space (Wikipedia, 2007b).

Kernel Type Selection: Process of selecting the form of the kernel function.

Outlier: Data objects from a database, which do not comply with the general behavior or model of the data (Han and Kamber, 2006). **Outlier detection and analysis** is an important data mining task, named **outlier mining**, with applications, for example, in fraud detection.

Score Distribution Model: Model associated with a particular combination of the distributions of the score values of relevant and non-relevant data. There are several models proposed so far, such as exponential-exponential, normal-normal, Gaussian-exponential, gamma-gamma, and so on. However, a good reason for choosing any particular model is based on how good the distributions fit a set of empirical data.

TREC: Text Retrieval Conference that focuses on different information retrieval research areas, or tracks, and provides the infrastructure necessary for large-scale evaluation of text retrieval methodologies.

TRECVID: A conference separate from the TREC conference that focuses on a list of different information retrieval (IR) research areas in content based retrieval of video.

Modeling the KDD Process

Vasudha Bhatnagar
University of Delhi, India

S. K. Gupta
IIT, Delhi, India

INTRODUCTION

Knowledge Discovery in Databases (KDD) is classically defined as the "nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in large databases" (Fayyad, Piatetsky-Shapiro & Smyth, 1996a). The recently developed KDD technology is based on a well-defined, multi-step "**KDD process**" for discovering knowledge from large data repositories. The basic problem addressed by the KDD process is one of mapping low-level data (operational in nature and too voluminous) to a more abstract form (descriptive approximation or model of the process that generated the data) or a useful form (for example, a predictive model) (Fayyad, Piatetsky-Shapiro & Smyth, 1996b). The KDD process evolves with pro-active intervention of the domain experts, data mining analyst and the end-users. It is a 'continuous' process in the sense that the results of the process may fuel new motivations for further discoveries (Chapman et al., 2000). Modeling and planning of the KDD process has been recognized as a new research field (John, 2000).

In this chapter we provide an introduction to the *process of knowledge discovery in databases (KDD process),* and present some models (conceptual as well as practical) to carry out the KDD endeavor.

BACKGROUND

The process of Knowledge Discovery in Databases consists of multiple steps, and is inherently iterative in nature. It requires human interaction for its applicability, which makes the process subjective. Various parameters require to be adjusted appropriately before the outcome of the process can be applied for decision making.

The process starts with the task of understanding the domain in the context of the goal of the endeavor, and ends with the task of interpretation and evaluation of the discovered patterns. Human centric nature of the process has been emphasized since the early days of inception of the KDD technology (Brachman & Anand, 1996) and vindicated by the veterans (Ankerst, M. 2002). The core of KDD process employs "data mining" algorithms, which aim at searching for interesting models and patterns in the vast search space, and are responsible for actually discovering nuggets of knowledge (Fayyad, Piatetsky-Shapiro & Smyth, 1996a). Often, the key to a successful KDD effort lies not so much in the act of data mining alone but in the pre and post mining steps of the process. Understanding the whole KDD process therefore becomes imperative for the success of the knowledge discovery exercise. Further, a formal model of the KDD process also helps in differentiating and comparing two or more approaches for KDD endeavor.

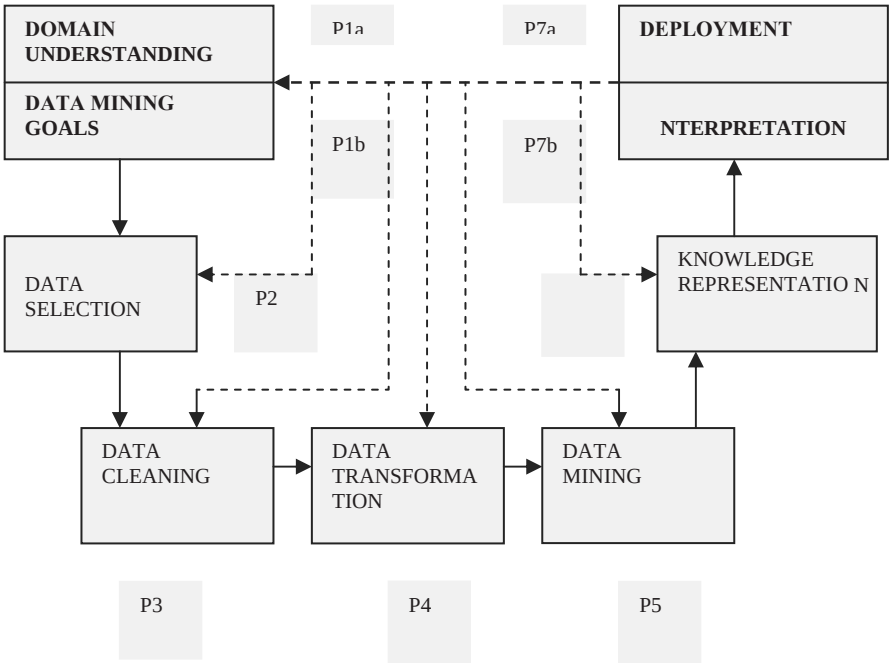
GENERIC STEPS OF THE KDD PROCESS

Figure 1 shows a simple model of the KDD process exhibiting the logical sequencing of the various process steps. The model allows the data miner to effortlessly map the logical process steps (P1 to P7) to the corresponding physical computing processes.

The data flows in a straight forward manner from each process step to the subsequent step as shown by solid lines. The dash lines show the control flow and indicate optional iteration of process steps after the discovered knowledge has been evaluated. We describe below the generic steps of a KDD process.

Domain Understanding and Defining Data Mining Requirements (Step P1(a, b)) are the tasks that

Figure 1. Steps of the KDD process



comprise the first step of the KDD process. Identifying and defining the project goals, assessing the feasibility, planning and preparation of subsequent steps, are some of the sub-tasks that need to be performed during this step. Understanding of the basic characteristics of available data and knowledge of the domain, play an important role in crystallizing the project objectives. Experience and expertise of the domain experts, decision-makers and the end users help in translating the project objectives into knowledge discovery goals.

Data Selection (Step P2) involves laying down the criteria for including the data to be analyzed or excluding the unwanted data. Since all the data may not be relevant for the knowledge discovery goals, the data mining analyst selectively identifies the relevant data, depending on the business questions that need to be answered. For instance, in order to explore the buying patterns of the customers of a particular geographical area, customer addresses can be the selection criterion.

Data Cleaning (Step P3) is performed to ensure domain and semantic validity of data with respect to the goals set in Step P1b. Faulty data capturing methods, transmission errors or legal issues are some causes of "noisy data" in a database apart from the practical data gathering pitfalls. Strategies to handle missing and noisy data have to be decided by the end user and/or domain expert, and must be implemented faithfully before applying the mining algorithm.

Data cleaning assumes added significance for the successful outcome of the KDD process, since most of the mining algorithms do not address the problem of dirty or missing data.

Data Transformation (Step P4) is often required because of syntactic reasons. For example, in a distributed database if the salary attribute is stored in different currencies then it has to be transformed to a pre-decided common currency before undertaking mining. Some data mining algorithms require data to be in a specific format. Transforming numerical attributes to categorical attributes is commonly required before applying *clustering* or *classification* algorithms. For *multilevel association rule* mining, data transformations are required to facilitate mining at different levels of abstraction.

Data Mining (Step P5) results into discovery of hidden patterns by applying a suitable mining algorithm over the "*pre-processed*" (selected, cleaned and transformed) data. The choice of the mining algorithm is influenced by the specification of the discovery goals, type of knowledge to be discovered and nature of data. The mining analyst may have to re-engineer an algo-

rithm in case no suitable algorithm exists to meet the requirement. Scrupulous execution of the earlier steps of the KDD process significantly improves the quality of patterns discovered by the mining algorithm.

Presentation of the Discovered Knowledge (Step P6) facilitates the interpretation of the mining results. Recent advances made in the area of visualization of discovered knowledge make presentation of discovered knowledge more intuitive and meaningful. State-of-art visualization tools for displaying the discovered knowledge to the user are available in most of the data mining packages.

Interpretation and Deployment of the Discovered Knowledge (Step P7(a, b)) are the final steps in an iteration of the KDD process. These steps are crucial for the flow of control, since they involve evaluation of the discovered patterns, either by analysis or by their deployment, and deciding the next action. If the discovered patterns are useful in the context of the mining goals, the KDD project is considered to be successful. Otherwise there may be a need to refine or redefine the goals, or take another look at each of the steps in the KDD process and make necessary changes in steps P2 – P6. This makes KDD an *iterative* and *interactive* process.

It is worth noting that process steps P1 and P7 require intense involvement of the mining analyst, domain expert and the end user - the human elements in the knowledge discovery process.

KDD PROCESS MODELS

The task of modeling the KDD process is challenging because of the diversity of the processes and their uniqueness with respect to each application. In actual practice even within a single domain, the details of the process may vary for each application. We give a brief account of some popular models.

William's Model

A four element - four stage model of the KDD process has been proposed by William and Huang (1996). Being a *conceptual model*, it attempts to capture elements of the process in an abstract way. Motivation for such a model comes from the necessity of a formal theory to guide the practitioners in the effective use of KDD technology.

William's model uses a 4-tuple {*D, L, F, S*} to identify necessary and sufficient elements of the KDD process. Here *D* is the target database, *L* is the knowledge representation language for expressing the discovered model, *F* is the pattern evaluation function and *S* is the set of operations performing the knowledge discovery. *S* is a four stage, operative process element shown in Figure 2, which subsumes the traditional KDD process model (Figure 1). Summarization of several lower level steps in the traditional model to lesser number of higher-level steps does not undermine the complex iterative nature of the KDD process, and importance of human interaction during the process.

William's model provides a framework for distinguishing between different KDD approaches in terms of their constituent elements. Interestingly the process steps that support the KDD operations, themselves form an element in this model.

Reinartz's Model

A practical KDD process model based on the overview of the life cycle of KDD projects at a generic level is shown in Figure 3 (Reinartz, 1997). The model permits need based data flow and control flow in both directions and exhibits an ideal sequence of seven generic tasks in a KDD project.

Reinartz's model combines the tasks of data selection, cleaning and transformation as a single data preparation task. The next task is data exploration, which helps in understanding the interesting data characteristics using statistical and/or visualization tools. For example, identifying areas of high density in the data space using a visualization tool can give an idea of the probable clusters. In this model the interrelationship between the process steps is substantially improved. The transitions between different steps provide a structured and smooth road to go back to any of the previous process step at any stage during the KDD process. The explicit presentation of the transitions emphasizes the importance of evaluation (of the outcome of the current step) and decision-making (what next?) at the end of each process step. Involvement of human elements in the KDD process is reinforced effectively in this model.

Figure 2. The fourth element of the Williams' Model

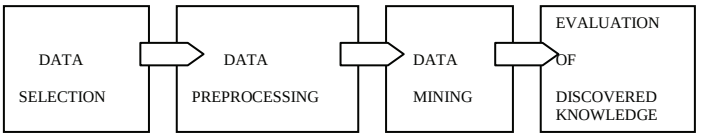
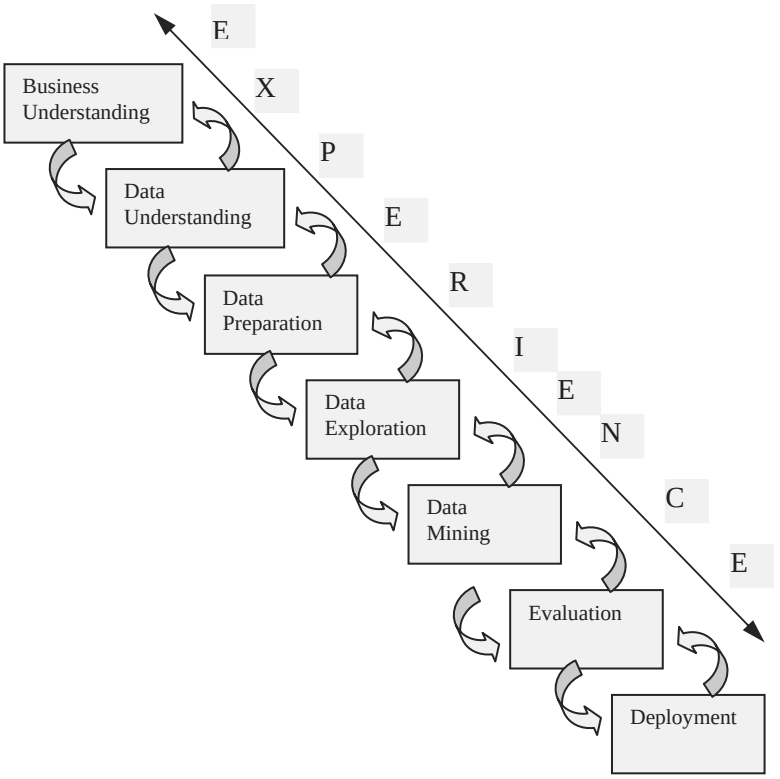


Figure 3. Life cycle of a KDD project (Reinartz 1997)

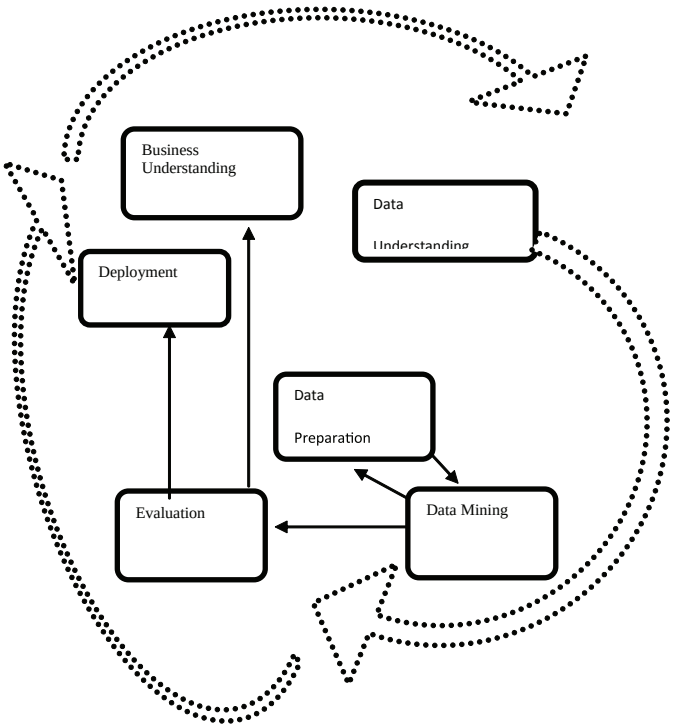


CRISP-DM Model

CRISP-DM (CRoss-Industry Standard Process for Data Mining) is an industry initiative, which aims to develop, validate and promote a standard data mining methodology for knowledge discovery in databases (Chapman et al., 2000). The CRISP-DM model shown in Figure 4 was developed by an industry consortium and it advocates a data mining methodology based on a formal KDD process. The methodology is non-proprietary and freely available. The proposed data mining methodology consists of tasks described at four level of abstraction (from general to specific) - phase, generic tasks, specialized tasks and process instances. We however restrict our attention to the underlying model on which the CRISP-DM methodology is based.

Figure 4 shows the CRISP-DM model, which offers systematic understanding of step-by-step direction, tasks and objectives for every stage of the process - from business understanding, data understanding, data preparation, modeling, to evaluation and deployment. The straight block arrows indicate some of the important and frequent dependencies between stages, though more such dependencies can arise in practice. Back and forth transitions may be unpredictable, complex and unique for each KDD endeavor. The circular arrows in the figure symbolize the continuous nature of the KDD process, indicating its continuity even after a solution has been deployed. It is premised that lessons learned during the process can trigger new, often more focused business questions. Continuity of the KDD process also

Figure 4. CRISP-DM methodology for KDD



ensures that the subsequent KDD exercises will benefit from the experiences of previous ones.

The CRISP-DM model is highly demanding in terms of user interaction and considers the discovery process to be continuous in nature. These features of this model are important since they distinguish it from the earlier models. Due attention is given to the need based transitions from one stage to another, emphasizing the need for critical and decisive evaluation of the outcome at the end of each stage. Although the model is not based on theoretical or academic principles, yet it is very successful because of its neutrality with respect to industry, tools and applications.

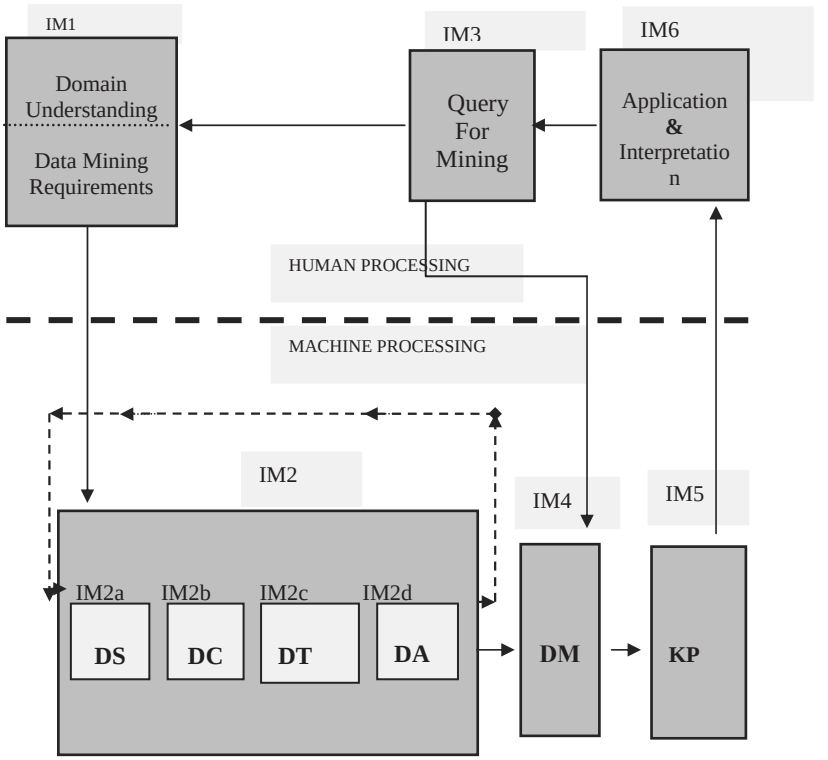
I-MIN Model

The I-MIN model for KDD Process (Figure 5) is based on *Intension Mining* scheme in which the mining requirements are formally documented as schema (Bhatnagar, 2001). This model inherently supports experimentation and monitoring, both functionalities desirable in business and scientific data explorations. Mining queries can be formulated and applications can be developed for experimentation in the KDD process, by varying the target dataset and the mining parameters. The model is downward compatible and provides full functionality available from the traditional KDD process model. Designing and integrating the agents for each of the process steps can realize the model

Step IM1 corresponds to step P1 of the traditional KDD process model (Figure 1) and results into design of a knowledge discovery schema, which formally documents the knowledge discovery goals. The schema is compiled and the resulting meta-data is stored for future use during subsequent steps of the process.

Step IM2 is a compound step in which steps (IM2a - IM2c) are data preparation tasks that map to steps (P2 - P4) of traditional KDD process model. Step IM2d is novel and is responsible for aggregation and possibly partial data analysis, which is carried out in step P5 of traditional KDD process model. Aggregation results into synopsis in a format specific for a mining algorithm. Since the functions for pre-mining and aggregation are already specified in schema, they are performed

Figure 5. I-MIN model for KDD process (DS : Data Selection, DC : Data Cleaning, DT : Data Transformation, DA : Data Aggregation, DM : Data Mining, KP : Knowledge Presentation)



without any human intervention. The dotted line around IM2 (with arrows) indicates that this compound step is periodically and automatically repeated. On each invocation the incremental database is processed and a corresponding aggregated knowledge structure (synopsis) is produced.

Step IM3 signifies initiation of the actual data mining phase. This user-initiated step commences with either formulation of mining query or execution of an application. During step IM4, the query or application is vetted against the corresponding schema and mining algorithm specified in the schema is invoked. The actual mining takes place over the synopsis aggregated during Step IM2d. Step IM4 together with step IM2d corresponds to step P5 of traditional KDD process model. Repeated querying on aggregated data with different sets of parameters facilitates experimentation. Designing applications to analyze the aggregates generated during Step IM2d, helps in monitoring the changing data characteristics in an evolving database.

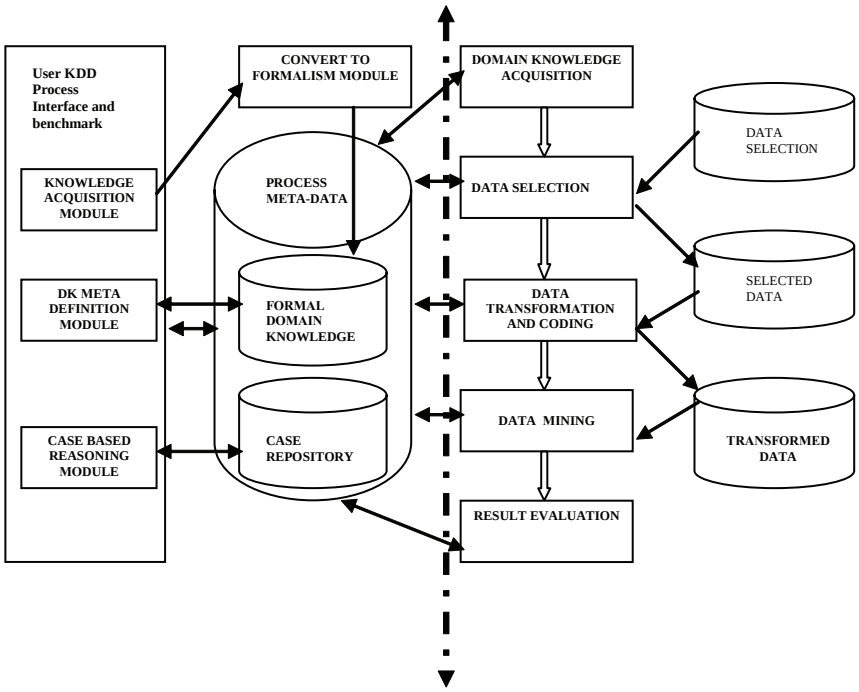
The resulting knowledge is presented and interpreted/applied in steps (IM5 - IM6) as in (P6 - P7). The arrow from IM6 to IM3 permits experimentation and monitoring, useful in real life KDD endeavors, where answers to the business questions set as goal, often fuel the need to answer more questions.

Redpath's Model

Successful automation of the KDD process is a challenging task because of the multiplicity of approaches that often need to be combined for each endeavor. Redpath & Srinivasan (2004) have argued that capturing domain knowledge before starting the endeavor helps in reducing the complexity of the process and facilitates automation. Architecture for partial automation of the KDD process is shown in Figure 6. The architecture is based on the premise that domain knowledge can be exploited to assist in most of the steps of the KDD process including data selection, algorithm selection, results evaluation and results interpretation. This model also utilizes elements like Case Based Reasoning, Process Metadata to optimize the partially automated KDD process.

To the left of the central line lies the process of acquisition of domain knowledge and on the right the

Figure 6. Architecture for a partially automated KDD system (Redpath & Srinivasan, 2004)



KDD process is shown. The arrows across the central line show the role of domain knowledge in executing various KDD process steps automatically.

FUTURE TRENDS

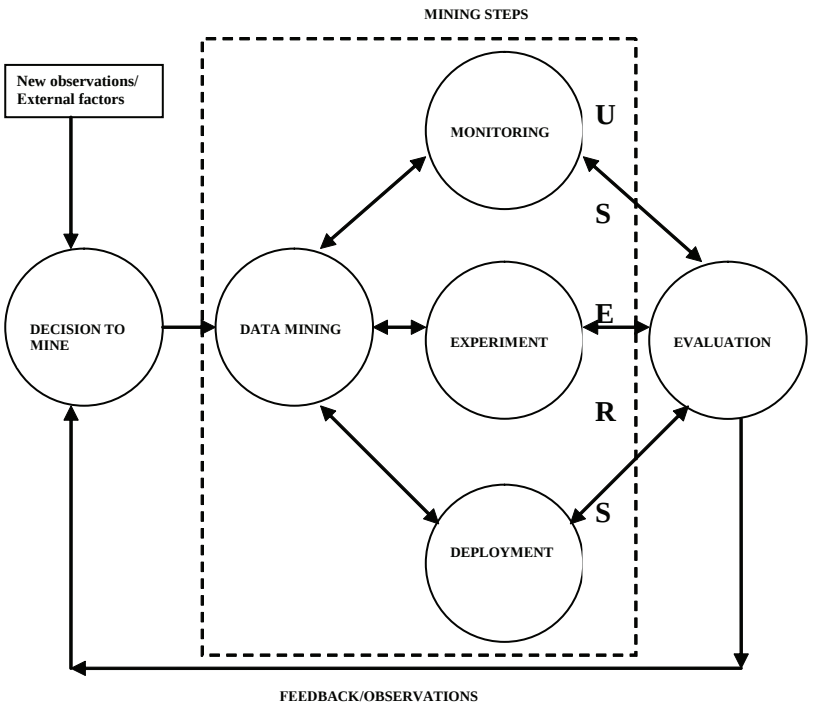
In practice, discovery of knowledge fuels needs to discover more knowledge with varying perspectives. Growth in demands for new mining applications, particularly in environments with evolving database, can be satisfied by either designing a new KDD process for each requirement, or by modeling the process in such a manner that it can meet multiple logically related requirements. It is envisaged that an organization would have several KDD endeavors running either concurrently or at different points in time, for different needs. A need to control, monitor and audit these endeavors would naturally emerge in a big way. Consequently, issues of security, privacy and proper accounting would need to be addressed. A complete and structured documentation of all KDD operations in an organization would help to some extent.

Figure 7 summarizes the real life needs in a KDD savvy organization. KDD process models that satisfy these needs and have a sound theoretical background need to be developed for deeper penetration of KDD technology in decision sciences. Designing new mining algorithms that can be used in I-MIN model assumes importance, since I-MIN model meets most of the functionalities desired in the real life KDD projects.

CONCLUSION

Knowledge Discovery in Databases is a complex human centric process requiring different tools. The success of a KDD project depends on a good mix of tools and skilled analyst. Several models have been proposed to capture the real life KDD endeavors. All

Figure 7. Desired functionality in KDD projects (Bhatnagar, 2001)



the models are unanimous as far as the process steps and data flow are concerned. The differences in the control flow set the models apart. The complex and unique control flow of each endeavor has retarded attempts for full automation of the KDD process. A mechanism to support knowledge discovery operations by allowing interactive explorations, experimentation and monitoring during the mining step is required for a truly user-centric KDD system to facilitate flexibility, efficiency and applicability.

REFERENCES

Ankerst, M. (2002). Report on the SIGKDD-2002 Panel: The Perfect Data Mining Tool: Interactive or Automated*?, SIGKDD Exploration. 4(2);* 110-111.

Bhatnagar V. (2001). Intension Mining – A New Paradigm for Knowledge Discovery in Databases. PhD Thesis, JMI, New Delhi, India.

Brachman, R.J., Anand, T. (1996). The Process of Knowledge Discovery in Databases: A Human-Centered Approach. In Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., & Uthurasamy, R. (eds.). *Advances in Knowledge Discovery and Data Minin*g. AAAI /MIT Press

Chapman, P., et al. (2000) . *CRISP-DM 1.0 Step-by-step data mining guide*. Technical Report. *www.crisp-dm.org ,CRISP-DM consortium.*

Fayyad, U. M., Piatetsky-Shapiro, G. & Smyth, P. (1996a). From Data Mining to Knowledge Discovery: An Overview. In Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., & Uthurasamy, R. (eds.). *Advances in Knowledge Discovery and Data Minin*g. AAAI /MIT Press

Fayyad, U. M., Piatetsky-Shapiro, G. & Smyth, P. (1996b). From Data Mining to Knowledge Discovery in Databases. *Artificial Intelligence Magazine.* Fall 1996, 37-54

Gupta S.K., Bhatnagar V. & Wasan S. K. (2000). User Centric Mining of Association Rules. *In Workshop on Data Mining, Decision Support, Meta Learning and ILP. PKDD 2000.* Lyons, France, September 13-16.

Gupta S.K., Bhatnagar V. & Wasan S. K. (2005). Architecture for Knowledge Discovery and Knowledge Management**.** *KAIS*, 7(3);310-336.

John, G., (1997). Enhancements to the Data Mining Process. PhD Thesis. Computer Science Department, Stanford University, USA.

Redpath R. & Srinivasan B. (2004). A Model for Domain Centered Knowledge Discovery in Databases. *In Proceedings of the IEEE 4th International Conference On Intelligent Systems Design and Application*, Budapest, Hungary.

Reinartz T. P. (1999). Focusing Solutions for Data Mining: Analytical Studies and Experimental Results in Real-World Domain. LNAI 1623, Springer-Verlag.

Williams G. J., & Huang Z. (1996). Modeling the KDD Process. Technical Report TR-DM-96013, CSIRO Division of Information Technology, Australia.

KEY TERMS

Domain Knowledge: The knowledge about the environment in which the KDD endeavor is to be carried out, acquired by the past experiences of the users/experts.

Data Mining: Step in the KDD process that invokes a data-mining algorithm and serves to actually discover patterns that are hidden within the data.

Intension Mining: Is a scheme of knowledge discovery, where the mining needs are documented as *Knowledge Discovery Schema*. The Intension Mining based system pre-processes the incoming data as it arrives in the database at periodic intervals and accumulates partial knowledge. These aggregates (knowledge concentrates) can be mined by the end-user by formulating a suitable query in a specific language (Bhatnagar, 2001).

Interestingness: It is an overall measure of the value of a pattern to the user. It is a combination of the measures of validity, novelty, usefulness, and simplicity of the discovered pattern and is subjective with respect to time and user.

KDD Process: The nontrivial process of discovering hidden knowledge from large data repositories.

Knowledge Discovery in Databases (KDD): Extraction of novel, valid, interesting and useful trends and patterns (nuggets of knowledge) from large data repositories.

Pattern: A pattern is a form/template/model (or, more abstractly, a set of rules) which can be used to make or to generate things or parts of a thing.

A Multi-Agent System for Handling Adaptive E-Services

Pasquale De Meo
Università degli Studi Mediterranea di Reggio Calabria, Italy

Giovanni Quattrone
Università degli Studi Mediterranea di Reggio Calabria, Italy

Giorgio Terracina
Università degli Studi Della Calabria, Italy

Domenico Ursino
Università degli Studi Mediterranea di Reggio Calabria, Italy

INTRODUCTION

An Electronic-Service (E-Service) can be defined as a collection of network-resident software programs that collaborate for supporting users in both accessing and selecting data and services of their interest present in a provider site. Examples of e-services are e-commerce, e-learning, e-government, e-recruitment and e-health applications. E-Services are undoubtely one of the engines presently supporting the Internet Revolution. Indeed, nowadays, a large number and a great variety of providers offer their services also or exclusively via the Internet.

BACKGROUND

In spite of their spectacular development and present relevance, E-Services are far to be considered a stable technology and various improvements could be thought for them. Many of the present suggestions for improving them are based on the concept of adaptivity, i.e., on the capability to make them more flexible in such a way as to adapt their offers and behaviour to the "environment" they are operating in. In this context, systems capable of constructing, maintaining and exploiting suitable profiles for users accessing E-Services appear capable of playing a key role in the future (Kobsa, 2007).

Both in the past and in the present, various E-Service providers exploit (usually rough) user profiles for proposing personalized offers. However, in most cases, the profile construction methodology adopted by them presents some problems. In fact, it often requires a user to spend a certain amount of time for constructing and updating his profile; in addition, the profile of a user stores only information about the proposals which he claims to be interested in, without considering other ones, somehow related to those just provided, possibly interesting him in the future and that he disregarded to take into account in the past.

In spite of present user profile handlers, generally, when accessing an E-Service, a user must personally search the proposals of his interest through it. We argue that, for improving the effectiveness of E-Services, it is necessary to increase the interaction between the provider and the user, on one hand, and to construct a rich profile of the user, taking his interests, needs and past behaviour into account, on the other hand.

In addition, a further important factor must be taken into account. Nowadays, electronic and telecommunications technology is rapidly evolving in such a way as to allow cell phones, palmtops and wireless PDAs to navigate on the Web. These mobile devices do not have the same display or bandwidth capabilities as their desktop counterparts; nonetheless, present E-Service providers deliver the same contents to all device typologies (Communications of the ACM, 2002; ; Smith, Cotter & Oman, 2007).

In the past, various approaches have been proposed for handling E-Service activities; some of them are agent-based. As an example:

- In (Anand, Kearney & Shapcott, 2007) an approach to helping users looking for relevant items

is described. In order to generate its recommendations, this approach integrates user ratings with an ontology describing involved items. This approach is particularly suited for the e-commerce domain.

- In (Mahmood & Ricci, 2007) a travel recommender system, based on the Intelligent Agent technology, is presented. This system builds user profiles by exploiting Reinforcement Learning techniques, and models the recommendation process as a Markov Decision Process.
- In (Medjahed & Bouguettaya, 2005) the Authors propose WebSenior, a system using ontologies to automatically generate Web Services customized to senior citizen needs and government laws. WebSenior is able to manage both simple and composite services. In order to generate these last, it defines a graph, called dependency diagram, in which each vertex is associated with a simple service and each edge denotes a dependency relationship between a pair of services. The problem of computing composite services is, then, regarded as the problem of computing paths in the dependency diagram; this last problem is solved by applying the Floyd-Warshall dynamic programming algorithm.
- In (Ahn, Brusilovsky, Grady, He & Syn, 2007) YourNews, a system capable of helping users in accessing news located on the Web, is proposed. YourNews relies on a user profile built by unobtrusively monitoring user behaviour; this profile is open, in the sense that the corresponding user can interactively provide feedbacks that will be exploited by YourNews to enhance its accuracy.
- In (De Meo, Quattrone, Terracina & Ursino, 2007) a XML-based multi-agent recommender system for supporting online recruitment services is presented. This system handles user profiles for supporting a personalized job search over the Internet. In order to perform its recommendations, it exploits some advanced techniques (e.g., least square error and Pavlovian learning).
- In (Fang & Sheng, 2005) ServiceFinder, a system conceived for supporting citizens in their selection of relevant e-government services, is proposed. ServiceFinder uses Web Mining techniques to discover the *N* services best matching user needs and modifies the home page of an institutional e-government portal by adding to it *N* hyperlinks pointing to these services.
- In (Srivihok & Sukonmanee, 2005) a system capable of supporting e-tourism activities is proposed. This system analyzes past user behaviours and applies the Q-Learning algorithm to build a user profile. After this, it applies a reinforcement algorithm on both user and trip profiles in such a way as to associate a score with each trip proposal. These last are, then, ranked on the basis of their scores and only the top five are presented to the user.

All these systems construct, maintain and use rich data structures regarding both user needs and behaviours; therefore, we can consider them adaptive w.r.t. the user; however, none of them is adaptive w.r.t. the device.

On the other side, in many areas of computer science research, a large variety of approaches that adapt their behaviour on the basis of the device the user is exploiting, has been proposed. As an example:

- In (Samaras & Panayiotou, 2004) the system mPERSONA, aiming to support users equipped with wireless devices to access information sources located on the Web, is proposed. mPERSONA relies on a mobile multi-agent architecture; it associates a user profile (consisting of a set of keywords) with each user and represents the contents of an information source as a hierarchy (called metadata tree). Each time a user submits a query to an information source, mPERSONA isolates the portion of hierarchy (and the corresponding information objects) best fitting his requests; after this, it considers the features of the device he is currently exploiting and adapts the selected contents to them.
- In (Lee, Kang, Choi & Yang, 2006) an approach to Web content adaptation for mobile users is presented. This approach stores user preferences in a suitable profile. When a user is involved in Web browsing activities, it partitions a Web page into blocks, filters out those judged unnecessary and sorts the other ones on the basis of their relevance to the user. Finally, it presents sorted blocks to the user.

- In (Muntean & McManis, 2006) an approach to supporting mobile users in their browsing activities is proposed. This approach considers both user preferences (encoded in a user profile) and device features to suitably modify the content of the Web pages visited by the user; for instance, it can compress or eliminate images, if their sizes are excessive.
- In (Wei, Bhandarkar & Li, 2007) an approach to delivering multimedia data, taking both user preferences and device constraints into account, is proposed. This approach estimates how much a multimedia information object fits both the exigencies of a user and the constraints of the device he is currently exploiting. This information is used to define a knapsack-like problem whose solution allows the selection of the objects to deliver.

These approaches are particularly general and interesting; however, none of them has been conceived for handling E-Services.

MAIN THRUST OF THE CHAPTER

Challenges to Face

In order to overcome the problems outlined above, some challenges must be tackled.

Firstly, a user can access many E-Services, operating in the same or in different application contexts; a faithful and complete profile of him can be constructed only by taking his behaviour on accessing all these sites into account, and by storing the corresponding profile in a unique data structure on his side.

Secondly, for a given user and E-Service provider, it should be possible to compare the profile of the user with the offers of the provider for extracting those proposals that probably will interest him. Existing techniques for satisfying such a requirement are mainly based on either log files or cookies. In the former case they cannot match user preferences and E-Service proposals; in the latter one they need to know and exploit some personal information that a user might consider private.

Thirdly, the typical "one-size-fits-all" philosophy of present E-Service providers should be overcome by developing systems capable of adapting their behaviour to both the profile of the user and the characteristics of the device he is exploiting for accessing them (Communications of the ACM, 2002).

Fourthly, relationships and interactions among users should be taken into account; in fact, they can be exploited to guarantee a certain proactivity degree to the E-Service provider, on one hand, and to create communities of citizens showing similar needs and desires, on the other hand.

System Description

The system we present in this chapter (called *E-Service Adaptive Manager*, ESA-Manager) aims to tackle all challenges mentioned above. It is a multi-agent system for handling user accesses to E-Services, capable of adapting its behaviour to both user and device profiles.

In ESA-Manager a *Service Provider Agent* is present for each E-Service provider, handling the proposals stored therein, as well as the interaction with users. In addition, an agent is associated with each user, adapting its behaviour to the profiles of both the user and the device he is currently exploiting. Actually, since a user can access E-Service providers by means of different devices, his profile cannot be stored in only one of them; indeed, it must be handled and stored in a support different from the devices generally exploited by him for accessing E-Service providers. As a consequence, it appears compulsory the exploitation of a *Profile Agent*, storing the profiles of both involved users and devices, and a *User-Device Agent*, associated with a specific user operating by means of a specific device, supporting him in his activities. As a consequence of the previous reasoning, for each user, a unique profile is mined and maintained, storing information about his behaviour in accessing all E-Service providers. In this way, ESA-Manager tackles the first challenge mentioned above.

Whenever a user accesses an E-Service by means of a certain device, the corresponding Service Provider Agent sends information about its proposals to the User-Device Agent associated with him and the device he is exploiting. The User-Device Agent determines similarities among the proposals presented by the provider and the interests of the user. For each of these similarities, the Service Provider Agent and the User-Device Agent cooperate to present to the user a group of Web

pages, adapted to the exploited device, illustrating the proposal. We argue that this behaviour provides ESA-Manager with the capability of supporting the user in the search of proposals of his interest delivered by the provider. In addition, the algorithms underlying ESA-Manager allow it to identify not only the proposals probably of interest to him in the present but also other ones possibly of interest to him in the future and that he disregarded to take into account in the past. In our opinion, this is a particularly interesting feature for a new approach devoted to deal with E-Services. Last, but not the least, it is worth observing that, since user profile management is carried out at the user side, no information about the user profile is sent to E-Service providers. In this way, ESA-Manager solves privacy problems left open by cookies. All reasonings presented above show that ESA-Manager is capable of tackling also the second challenge mentioned previously.

In ESA-Manager device profile plays a central role. Indeed, the proposals of a provider shown to a user, as well as their presentation formats, depend on the characteristics of the device the user is presently exploiting. However, the ESA-Manager capability of adapting its behaviour to the device the user is exploiting is not restricted to the presentation format of the proposals; in fact, exploited device can influence also the computation of the *interest degree* associated with the proposals presented by each provider. Specifically, one of the parameters which the interest degree associated with a proposal is based on, is the time the user spends in visiting the corresponding Web pages. In ESA-Manager, this time is not considered as an absolute measure, but it is normalized w.r.t. both the characteristics of the exploited device and the navigation costs. This reasoning allows us to argue that ESA-Manager tackles also the third challenge mentioned above.

In addition, ESA-Manager uses a Social Network to partition citizens accessing the system into homogeneous clusters called *communities*; these communities are set up on the basis of shared interests/needs of their members rather than on demographic data. Each time a citizen submits a query, ESA-Manager forwards the corresponding answers (i.e., the corresponding service proposals) also to the members of his community.

All these activities are carried out by a *Social Network Agent* which catches user queries and exploits them to both organize communities and handle communications with them. This confers a high level of *proactivity* to our system because it can recognize and recommend potentially relevant services to a citizen even though he is not aware of their existence. In addition, if, for a citizen, interests/needs have been changed, the Social Network Agent automatically assigns him to another community and, at the same time, suggests him the latest services recommended to the members of his new community. In this way, our system tackles also the fourth challenge mentioned above.

A final important characteristic of our system is that it encourages a socially inspired mechanism for service management. In fact, users can freely discuss among them to propose new services; as a consequence, the top-down approach for service delivery (in which providers push their services to users and impose them their formalism, terminology, access policy, features, etc.) is coupled and, hopefully, substituted by a bottom-up one (in which users join up to form communities who raise their requirements about desired services to providers). This provides our system with notable social capabilities in that it can suitably channelize user demands. In fact, in this scenario, users are encouraged to debate in such a way as to propose the activation of new services of interest to them.

FUTURE TRENDS

The spectacular growth of the Internet during the last decade has strongly conditioned the E-Service landscape. Such a growth is particularly surprising in some application domains, such as financial services or e-government.

As an example, the Internet technology has enabled the expansion of financial services by integrating the already existing, quite variegate, financial data and services and by providing new channels for information delivery. However, E-Services are not a leading paradigm only in business contexts; for instance, they are vigorously applied by governmental units at national, regional and local levels around the world.

A further interesting research trend consists of investigating the possibility to integrate an E-Service access system with a Decision Support one; in this way, user behaviour could be analyzed (for instance, by means of a Data Mining tool) for determining the key features of the most appreciated services. This information could be particularly useful for provider managers when they need to decide the new services to propose. This last feature would allow the realization of a hybrid E-

Service system, embodying the functionalities of both seeker-oriented and company-oriented systems.

CONCLUSION

In this paper we have proposed ESA-Manager, an adaptive multi-agent system for supporting a user, accessing an E-Service provider, in the search of proposals appearing to be appealing according to his interests and behaviour.

We have shown that ESA-Manager is adaptive w.r.t. the profile of both the user and the device he is currently exploiting. Finally, we have seen that it is proactive in that it suggests potentially interesting services to a user even if he did not explicitly require them.

As for future work, we would like to investigate whether a hierarchical clustering algorithm can be fruitfully exploited to improve the quality of user community management. The output of this algorithm would be a tree-like data structure whose nodes represent potential communities. This data structure would allow specialization/generalization relationships among communities to be handled; in fact, nodes placed at the top of the hierarchy would represent wide communities of loosely linked users (e.g., the community of students), whereas nodes at the bottom would be associated with narrow communities of tightly linked users (e.g., the community of working students). This hierarchical community organization could be extremely useful in the proactive suggestion of services, carried out by our system. In fact, in order to find the citizen communities potentially interested to a service, our system could run across the hierarchy, starting from its root, in such a way as to find the widest community potentially interested to it.

REFERENCES

Ahn, J., Brusilovsky, P., Grady, J., He, D. & Syn, S.Y. (2007). Open user profiles for adaptive news systems: help or harm?. *Proc. of the International Conference on World Wide Web*, pages 11-20, Banff, Alberta, Canada. ACM Press.

Anand, S.S., Kearney, P. & Shapcott, M. (2007). Generating semantically enriched user profiles for Web personalization. *ACM Transactions on Internet Technologies* 7(4), Article N. 22.

Communications of the ACM (2002). *Adaptive Web*. Volume 45(5). ACM Press.

De Meo, P., Quattrone, G., Terracina, G. & Ursino, D. (2007). An XML-based Multi-Agent System for Supporting Online Recruitment Services. *IEEE Transactions on Systems, Man and Cybernetics*, 37(4), 464-480.

Dolog, P., Simon, B., Klobucar, T. & Nejdl, W. (2008). Personalizing Access to Learning Networks. *ACM Transactions on Internet Technologies.* 8(2), Forthcoming

Fang, X. & Sheng, O.R.L. (2005). Designing a better Web portal for digital government: a Web-mining based approach. *Proc. of the National Conference on Digital Government Research*, pages 277-278, Atlanta, Georgia, USA. Digital Government Research Center.

Kobsa A.(2007). Generic User Modeling Systems. In *The Adaptive Web, Methods and Strategies of Web Personalization.* P. Brusilovsky, A. Kobsa and W. Nejdl (Eds.), pages 136-154, Springer.

Lee, E., Kang, J., Choi, J. & Yang, J. (2006). Topic-Specific Web Content Adaptation to Mobile Devices. *Proc. of the IEEE/WIC/ACM International Conference on Web Intelligence*, pages 845-848, Hong Kong. IEEE Computer Society Press.

Mahmood, T. & Ricci, F. (2007). Learning and adaptivity in interactive recommender systems. *Proc. of the International Conference on Electronic Commerce*, pages 75-84, Minneapolis, Minnesota, USA. ACM Press.

Medjahed, B. & Bouguettaya, A. (2005). Customized delivery of E-Government Web Services. *IEEE Intelligent Systems*, 20(6), 77-84.

Muntean, C.H. & McManis, J. (2006). Fine grained content-based adaptation mechanism for providing high end-user quality of experience with adaptive hypermedia systems. *Proc. of the International Conference on World Wide Web*, pages 53-62, Edinburgh, Scotland, UK. ACM Press.

Samaras, G. & Panayiotou, C. (2004). mPERSONA: Personalized Portals for the Wireless User: An Agent Approach. *Mobile Networks and Applications*, 9(6), 663-677.

Smyth, B., Cotter, P. & Oman S. (2007). Enabling Intelligent Content Discovery on the Mobile Internet. *Proc. of the AAAI Conference on Artificial Intelligence*. pages 1744-1751, Vancouver, British Columbia, Canada, AAAI Press.

Srivihok, A. & Sukonmanee, P. (2005). E-commerce intelligent agent: personalization travel support agent using Q Learning. *Proc. of the IEEE International Conference on Electronic Commerce*, pages 287-292, Xian, China. ACM Press.

Wei, Y., Bhandarkar, S.M. & Li, K. (2007). Video personalization in resource-constrained multimedia environments. *Proc. of the International Conference on Multimedia*, pages 902-911, Augsburg, Germany. ACM Press.

KEY TERMS

Agent: A computational entity capable of both perceiving dynamic changes in the environment it is operating in and autonomously performing user delegated tasks, possibly by communicating and cooperating with other similar entities.

Agent Ontology: A description (like a formal specification of a program) of the concepts and relationships that can exist for an agent or a community of agents.

Adaptive System: A system adapting its behaviour on the basis of the environment it is operating in.

Device Profile: A model of a device storing information about both its costs and capabilities.

E-Service: A collection of network-resident software programs that collaborate for supporting users in both accessing and selecting data and services of their interest handled by a provider site.

Multi-Agent System (MAS): A loosely coupled network of software agents that interact to solve problems that are beyond the individual capacities or knowledge of each of them.

User Modeling: The process of gathering information specific to each user either explicitly or implicitly. This information is exploited in order to customize the content and the structure of a service to the user's specific and individual needs.

User Profile: A model of a user representing both his preferences and his behaviour.

Multiclass Molecular Classification

Chia Huey Ooi
Duke-NUS Graduate Medical School Singapore, Singapore

INTRODUCTION

Molecular classification involves the classification of samples into groups of biological phenotypes. Studies on molecular classification generally focus on cancer for the following reason: Molecular classification of tumor samples from patients into different molecular types or subtypes is vital for diagnosis, prognosis, and effective treatment of cancer (Slonim, Tamayo, Mesirov, Golub, and Lander, 2000). Traditionally, such classification relies on observations regarding the location (Slonim et al., 2000) and microscopic appearance of the cancerous cells (Garber et al., 2001). These methods have proven to be slow and ineffective; there is no way of predicting with reliable accuracy the progress of the disease, since tumors of similar appearance have been known to take different paths in the course of time.

With the advent of the microarray technology, data regarding the gene expression levels in each tumor sample may now prove to be a useful tool in molecular classification. This is because gene expression data provide snapshots of the activities within the cells and thus, the profile of the state of the cells in the tissue. The use of microarrays for gene expression profiling was first published in 1995 (Schena, Shalon, Davis, and Brown, 1995). In a typical microarray experiment, the expression levels of up to 10,000 or more genes are measured in each sample. The high-dimensionality of the data means that feature selection (FS) plays a crucial role in aiding the classification process by reducing the dimensionality of the input to the classification process. In the context of FS, the terms *gene* and *feature* will be used interchangeably in the context of gene expression data.

BACKGROUND

The objective of FS is to find from an overall set of N features, the subset of features, S, that gives the best classification accuracy. This feature subset is also known as the *predictor set*. There are two major types of FS techniques, filter-based and wrapper techniques. Filter-based techniques have several advantages over wrapper techniques:

a. Filter-based techniques are computationally less expensive than wrapper techniques.
b. Filter-based techniques are not classifier-specific; they can be used with any classifier of choice to predict the class of a new sample, whereas with wrapper-based techniques, the same classifier which has been used to form the predictor set must also be used to predict the class of a new sample. For instance, if a GA/SVM (wrapper) technique is used to form the predictor set, the SVM classifier (with the same classifier parameters, e.g., the same type of kernel) must then be used to predict the class of a new sample.
c. More importantly, unlike the typical 'black-box' trait of wrapper techniques, filter-based techniques provide a clear picture of why a certain feature subset is chosen as the predictor set through the use of scoring methods in which the inherent characteristics of the predictor set (and not just its prediction ability) are optimized.

Currently, filter-based FS techniques can be grouped into two categories: *rank-based selection* (Dudoit, Fridlyand, and Speed, 2002; Golub et al., 1999; Slonim et al., 2000; Su, Murali, Pavlovic, Schaffer, and Kasif, 2003; Takahashi & Honda, 2006; Tusher, Tibshirani, and Chu, 2001) and state-of-the-art *equal-priorities scoring methods* (Ding & Peng, 2005; Hall & Smith, 1998; Yu & Liu, 2004). This categorization is closely related to the two existing criteria used in filter-based FS techniques. The first criterion is called relevance – it indicates the ability of a gene in distinguishing among samples of different classes. The second criterion is called redundancy – it indicates the similarity between

pairs of genes in the predictor set. The aim of FS is to maximize the relevance of the genes in the predictor set and to minimize the redundancy between genes in the predictor set.

Rank-based selection methods use only relevance as the criterion when forming the predictor set. Each of the *N* genes in the dataset is first ranked based on a score which indicates how relevant the gene is (i.e., its ability to distinguish among different classes). The *P* top-ranked genes are then chosen as the members of the predictor set. The choice of the value *P* is often based on experience or some heuristics (Dudoit et al., 2002; Li, Zhang, and Ogihara, 2004).

Due to the need for fast and simple reduction of dimensionality for gene expression datasets, the most ample instances of existing filter-based FS techniques for molecular classification are those of the rank-based category. This is because rank-based techniques consider only one criterion in forming the predictor set, resulting in lower computational cost than more complex techniques where two criteria are considered. Compared to rank-based techniques, there are considerably fewer existing instances of equal-priorities scoring methods, which use two criteria in forming the predictor set: relevance and redundancy (Ding & Peng, 2005; Hall & Smith, 1998; Yu & Liu, 2004). More importantly, in these methods *equal priority* is assigned to each of the two criteria (relevance and redundancy), hence the term 'equal-priorities scoring methods'.

MAIN FOCUS

There are two areas of focus in this article. The first is the area of FS for gene expression data. The second is the area of molecular classification based on gene expression data.

In existing studies on filter-based FS, at most two criteria are considered in choosing the members of the predictor set: relevance and redundancy. Furthermore, even in studies where both relevance and redundancy are considered (Ding & Peng, 2005; Hall & Smith, 1998; Yu & Liu, 2004), both criteria are given *equal* weights or priorities. Based on a two-class example used in another study (Guyon & Elisseeff, 2003), we begin to ask the question if the two criteria should *always* be given equal priorities regardless of dataset characteristics, namely the number of classes. To find the answer, Ooi, Chetty, and Gondal (2004) introduced the concept of differential prioritization as a third criterion to be used in FS along with the two existing criteria of relevance and redundancy. The concept was then tested on various gene expression datasets (Ooi, Chetty, and Teng, 2006; Ooi, Chetty, and Teng, 2007b). Differential prioritization works better than existing criteria in FS by forming a predictor set which is most optimal for the particular number of classes in the FS problem (Ooi, Chetty, and Teng, 2007a).

In the area of molecular classification, there is a lack of formal approach for systematically combining the twin problems of FS and classification based on the decomposition paradigm used in each problem. A multiclass problem is a problem in which there are three or more classes. It can be *decomposed* into several two-class sub-problems. The number of derived two-class sub-problems will depend on the type of the *decomposition paradigm* used. The rationale for doing this is that the two-class problem is the most basic, and thus, the easiest of classification problems (divide-and-conquer strategy). Furthermore, many classifiers such as Support Vector Machine (SVM) (Vapnik, 1998) are originally devised for the two-class problem.

Predictor Set Scoring Method

A FS technique is made of two components: the predictor set scoring method (which evaluates the goodness of a candidate predictor set) and the search method (which searches the gene subset space for the predictor set based on the scoring method). The FS technique is wrapper-based when classifiers are invoked in the predictor set scoring method. Filter-based FS techniques, on the other hand, uses criteria which are not classifier-based in order to evaluate the goodness of the predictor set. The criteria are listed below:

1. **Relevance:** The relevance of a predictor set tells us how well the predictor set is able to distinguish among different classes. It is summarized in the form of the average of the correlation between a member of the predictor set and the target class vector, which, in turn, represents the relevance of the particular feature (Hall & Smith, 1998). The target class vector (consisting of class labels of the training samples) represents the target class concept. Relevance is to be maximized in the search for the predictor set.

2. **Redundancy:** The redundancy in a predictor set indicates the amount of repetitions or similarity in terms of the information conveyed by the members of the predictor set. To measure the redundancy in a predictor set *S*, one possible scheme is to use the measure of *direct redundancy*, or measure of *redundancy* for short. This measure is defined as the sum of the pairwise correlations from all unique pairwise combinations of the members of *S*, normalized by dividing by the square of the size of *S*, $|S|^2$. Redundancy is to be minimized in the search for the predictor set.
3. **Antiredundancy:** It is necessary to consider an alternative to the measure of redundancy in the predictor set. The existing method directly minimizes the sum of the correlation between genes in the predictor set by making this sum the denominator in the score for measuring the goodness of the predictor set (Ding & Peng, 2005). We observe that this can present the problem of singularity at near-minimum redundancy. To avert this problem, Ooi et al. (2004) proposed an alternative method. In this method, a measure called antiredundancy represents the sum of the *lack* of correlation between genes in the predictor set. Antiredundancy is a measure opposite to the measure of redundancy in terms of quality, and is to be maximized in the search for the predictor set.
4. **Degree of differential prioritization (DDP):** The DDP is a parameter (domain from 0 to 1) which controls the ratio of the priority of maximizing relevance to the priority of minimizing redundancy. The application of the DDP in FS compels the search method to prioritize the optimization of one criterion (either relevance or redundancy) at the cost of the optimization of the other criterion. Thus, unlike other existing correlation-based techniques, in the DDP-based FS technique, the optimizations of both elements of relevance and redundancy need *not* necessarily have equal priorities in the search for the predictor set (Ooi et al., 2006).

A predictor set found using larger value of the DDP has more features with strong relevance to the target class vector, but also more redundancy among these features. Conversely, a predictor set obtained using smaller value of the DDP contains less redundancy among its member features, but at the same time also has fewer features with strong relevance to the target class vector.

The optimal value of the DDP decreases in an exponential-like manner as the number of classes increases. For instance, minimizing redundancy is more important for a 14-class problem than it is for a 2-class problem. The optimal value of the DDP is not necessarily 0.5 (equal-priorities scoring methods) or 1 (rank-based selection). Therefore, a degree of freedom in adjusting the priorities between maximizing relevance and maximizing antiredundancy is necessary in order to produce the best predictor set. This degree of freedom is embodied by the DDP and should be adjusted based on the number of classes in the dataset.

Decomposition Paradigm and FS-CA Combinations

In theory, it is possible to decompose a multiclass problem for either or both FS and classification. (The decomposition of the classification problem is also called classifier aggregation. The term classification here indicates both the process of building a classifier and the process of using that classifier to predict the class of a sample.) However, in studies where FS is conducted prior to molecular classification, decomposition is not consistently implemented for both FS and classification.

In some studies, the type of decomposition is the same for both FS and classification, e.g., *undecomposed* FS with *undecomposed* classification (Dudoit et al., 2002), or decomposed FS with similarly decomposed classification (Ramaswamy et al., 2001; Yeang et al., 2001). In other cases, decomposition may only be implemented for classification but not for FS (Chai & Domeniconi, 2004; Ding & Peng, 2005; Li et al., 2004), or vice versa. We note that the occurrence of the former (*undecomposed* FS with decomposed classification) is quite frequent in multiclass molecular classification studies. In fact, to the best of our knowledge, no study on multiclass molecular classification has presented the opposite combination of paradigms (decomposed FS with *undecomposed* classification).

In existing studies (Chai & Domeniconi, 2004; Ding & Peng, 2005; Li et al., 2004), if the decomposition type differs between FS and classification, the multiclass problem is often decomposed in classification, but *undecomposed* in FS. This occurs even in those

studies which focus on FS instead of classifier techniques. One possible reason is that studies proposing new FS techniques prefer to first test their techniques in the undecomposed paradigm since undecomposed FS is the most basic and natural form of FS. Hence, decomposition in FS, with applications to multiclass molecular classification, has not been investigated as widely as either *undecomposed* FS or decomposed classification.

Overall, there has been a lack of a formal, systematic approach for combining the distinct problems of FS and classification in the context of problem decomposition for multiclass molecular classification. To solve this problem, the FS-CA system is devised (Ooi et al., 2007b) in which FS and classification are systematically combined based on the decomposition paradigms.

In this system, three decomposition paradigms are considered for FS: ACA (all-classes-at-once or non-decomposition), OVA (one-vs.-all), and PW (pairwise). For classification, three classifier aggregation methods are considered: SM (single machine or non-decomposition), OVA, and PW. The nine possible FS-CA combinations are: ACA-SM, ACA-OVA, ACA-PW, OVA-SM, OVA-OVA, OVA-PW, PW-SM, PW-OVA, and PW-PW. An X-Y combination combines an 'X' FS decomposition paradigm with a 'Y' classifier aggregation method. For instance, the PW-OVA combination combines PW-decomposed FS with OVA classifier aggregation.

Predictor Set Size Framework

In the predictor set size framework, a system of comparisons is devised by fixing, to a predetermined constant, one of these size factors:

a. predictor set size formed per FS sub-problem, $P_{sp,q}$,
b. average number of genes used to induce one component classifier, $P_{cc,b}$, and
c. number of total genes used in all component classifiers, P.

The purpose of this framework is to fairly compare classification performance across the nine FS-CA combinations, or more specifically, across the three types of problem decomposition used in FS (ACA, OVA, and PW). The framework is necessary due to the fact that FS with no decomposition (ACA) is expected to perform worse when compared to FS with decomposition (OVA and PW) using the simple criterion of fixing $P_{sp,q}$ to a constant for all FS-CA combinations.

This is because while the number of genes selected per FS sub-problem, $P_{sp,q}$, is the same for all three paradigms, the number of genes used in inducing all component classifiers, P, is often greater when FS with decomposition (OVA or PW) is used compared to when FS with no decomposition (ACA) is used. In fact, it is possible to simply conclude that the better performance of the former (FS with decomposition) is due to the bulk of genes used in the overall classifier induction procedure. At the same time, it is possible that the difference in the values of $P_{cc,b}$ (whether for a given value of $P_{sp,q}$ or of P) may contribute to the disparity in performance among the FS-CA combinations as well.

FUTURE TRENDS

Advancements in the microarray technology mean increases in the dimensionality of the gene expression data and consequently, growth in demands for a good FS technique. The number of samples and classes analyzed are also increasing as a result of the improvement in the microarray technology. This means an increase in the need for a combination of FS and classification technique which can deal efficiently with both high-dimensional and multiclass data.

CONCLUSION

Aside from the conventional criteria of relevance and redundancy, the balance between the priorities assigned to each of these criteria is an important element in determining the goodness of the predictor set, especially for multiclass problems. Classification problems with larger number of classes (e.g., 14) require more emphasis on minimizing redundancy than problems with smaller number of classes (e.g., two classes).

Together with the predictor set size framework, the system of FS-CA combinations is useful in providing a framework for comparing classification performance across different combinations of decomposition paradigms in FS and classification.

In this article, we have discussed our approaches to the problems of FS and classification in the context

of multiclass molecular classification based on gene expression data. However, the approaches presented here are applicable, without any modification, to any high-dimensional, multiclass feature selection/classification problems, regardless of the domain.

REFERENCES

Chai, H., & Domeniconi, C. (2004). An evaluation of gene selection methods for multi-class gene expression data classification, *Proc. 2nd European Workshop on Data Mining and Text Mining in Bioinformatics*, 3-10.

Ding, C., & Peng, H. (2005). Minimum redundancy feature selection from microarray gene expression data, *Journal of Bioinformatics and Computational Biology*, *3(2)*, 185–205.

Dudoit, S., Fridlyand, J., & Speed, T. (2002). Comparison of discrimination methods for the classification of tumors using gene expression data, *J. Am. Stat. Assoc.*, *97*, 77-87.

Garber, M.E., Troyanskaya, O.G., Schluens, K., Petersen, S., Thaesler, Z., Pacyna-Gengelbach, M., v.d.R., M., , Rosen, G.D., Perou, C.M., Whyte, R.I., Altman, R.B., Brown, P.O., Botstein, D., & Petersen, I. (2001). Diversity of gene expression in adenocarcinoma of the lung, *Proc. Natl. Acad. Sci. USA*, *98(24)*, 13784–13789.

Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., Bloomfield, C.D., & Lander, E.S. (1999). Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring, *Science*, *286*, 531-537.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection, *J. Machine Learning Research*, *3*, 1157-1182.

Hall, M.A., & Smith, L.A. (1998). Practical feature subset selection for machine learning, *Proc. 21st Australasian Computer Science Conf.*, 181-191.

Li, T., Zhang, C., & Ogihara, M. (2004). A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression, *Bioinformatics*, *20*, 2429-2437.

Ooi, C.H., Chetty, M., & Gondal, I. (2004). The role of feature redundancy in tumor classification, *Proc. Int. Conf. Bioinformatics and its Applications (ICBA'04), Advances in Bioinformatics and its Applications, Series in Mathematical Biology and Medicine 8*, 197-208.

Ooi, C.H., Chetty, M., & Teng, S.W. (2006). Differential prioritization between relevance and redundancy in correlation-based feature selection techniques for multiclass gene expression data, *BMC Bioinformatics*, *7*, 320.

Ooi, C.H., Chetty, M., & Teng, S.W. (2007a). Characteristics of predictor sets found using differential prioritization, *Algorithms for Molecular Biology*, *2*, 7.

Ooi, C.H., Chetty, M., & Teng, S.W. (2007b). Differential prioritization in feature selection and classifier aggregation for multiclass gene expression datasets, *Data Mining and Knowledge Discovery*, *14(3)*, 329–366.

Ramaswamy, S., Tamayo, P., Rifkin, R., Mukherjee, S., Yeang, C.H., Angelo, M., Ladd, C., Reich, M., Latulippe, E., Mesirov, J.P., Poggio, T., Gerald, W., Loda, M., Lander, E.S., & Golub, T.R. (2001). Multi-class cancer diagnosis using tumor gene expression signatures, *Proc. Natl. Acad. Sci. USA*, *98*, 15149-15154.

Schena, M., Shalon, D., Davis, R.W., & Brown, P.O. (1995). Quantitative monitoring of gene expression patterns with a complementary DNA microarray, *Science*, *270*, 467–470.

Slonim, D.K., Tamayo, P., Mesirov, J.P., Golub, T.R., & Lander, E.S. (2000). Class prediction and discovery using gene expression data, *RECOMB 2000*, 263–272.

Su, Y., Murali, T.M., Pavlovic, V., Schaffer, M., & Kasif, S. (2003). Rankgene: Identification of diagnostic genes based on expression data, *Bioinformatics*, *19*, 1578-1579.

Takahashi, H., & Honda, H. (2006). Modified signal-to-noise: A new simple and practical gene filtering approach based on the concept of projective adaptive resonance theory (part) filtering method, *Bioinformatics*, *22(13)*, 1662-1664.

Tusher, V.G., Tibshirani, R., & Chu, G. (2001). Significance analysis of microarrays applied to the ionizing radiation response, *Proc. Natl. Acad. Sci. USA*, *98(9)*, 5116–5121.

Vapnik, V.N. (1998). *Statistical learning theory*. John Wiley and Sons.

Yeang, C.-H., Ramaswamy, S., Tamayo, P., Mukherjee, S., Rifkin, R.M., Angelo, M., Reich, M., Lander, E., Mesirov, J., & Golub, T. (2001). Molecular classification of multiple tumor types, *Bioinformatics, 17*, S316–S322.

Yu, L., & Liu, H. (2004). Redundancy based feature selection for gene expression data, *Proc. 2004 ACM SIGKDD*, 737-742.

KEY TERMS

Antiredundancy: A measure of the *lack* of correlation between genes in the predictor set – it is a measure opposite to the measure of redundancy in terms of quality.

Differential Prioritization: A parameter which controls the ratio of the priority of maximizing relevance to the priority of minimizing redundancy.

Equal-Priorities Scoring Methods: FS techniques in which relevance and redundancy have equal priorities as criteria in evaluating the goodness of the predictor set.

FS-CA: A combination of a particular FS decomposition paradigm and a particular classifier aggregation method.

Predictor Set Score: Measure of goodness of the predictor set – it is to be maximized in the search for the predictor set.

Rank-Based Selection Methods: FS techniques in which relevance alone is used as the criterion in forming the predictor set.

Redundancy: An indicator of the amount of repetitions or similarity in terms of the information conveyed by the members of the predictor set.

Relevance: A measure of the ability of a predictor set or a feature to distinguish among different classes.

Multidimensional Modeling of Complex Data

Omar Boussaid
University of Lyon, France

Doulkifli Boukraa
University of Jijel, Algeria

INTRODUCTION

While the classical databases aimed in data managing within enterprises, data warehouses help them to analyze data in order to drive their activities (Inmon, 2005).

The data warehouses have proven their usefulness in the decision making process by presenting valuable data to the user and allowing him/her to analyze them online (Rafanelli, 2003). Current data warehouse and OLAP tools deal, for their most part, with numerical data which is structured usually using the relational model. Therefore, considerable amounts of unstructured or semi-structured data are left unexploited. We qualify such data as "complex data" because they originate in different sources; have multiple forms, and have complex relationships amongst them.

Warehousing and exploiting such data raise many issues. In particular, modeling a complex data warehouse using the traditional star schema is no longer adequate because of many reasons (Boussaïd, Ben Messaoud, Choquet, & Anthoard, 2006; Ravat, Teste, Tournier, & Zurfluh, 2007b). First, the complex structure of data needs to be preserved rather than to be structured linearly as a set of attributes. Secondly, we need to preserve and exploit the relationships that exist between data when performing the analysis. Finally, a need may occur to operate new aggregation modes (Ben Messaoud, Boussaïd, & Loudcher, 2006; Ravat, Teste, Tournier, & Zurfluh, 2007a) that are based on textual rather than on numerical data.

The design and modeling of decision support systems based on complex data is a very exciting scientific challenge (Pedersen & Jensen, 1999; Jones & Song, 2005; Luján-Mora, Trujillo, & Song; 2006). Particularly, modeling a complex data warehouse at the conceptual level then at a logical level are not straightforward activities. Little work has been done regarding these activities.

At the conceptual level, most of the proposed models are object-oriented (Ravat et al, 2007a; Nassis, Rajugan, Dillon, & Rahayu 2004) and some of them make use of UML as a notation language. At the logical level, XML has been used in many models because of its adequacy for modeling both structured and semi structured data (Pokorný, 2001; Baril & Bellahsène, 2003; Boussaïd et al., 2006).

In this chapter, we propose an approach of multidimensional modeling of complex data at both the conceptual and logical levels. Our conceptual model answers some modeling requirements that we believe not fulfilled by the current models. These modeling requirements are exemplified by the Digital Bibliography & Library Project case study (DBLP)[1].

BACKGROUND

The DBLP (Ley & Reuther, 2006) represents a huge tank of data whose usefulness can be extended from simple reference searching to publication analysis, for instance:

- Listing the publications ordered by the number of their authors, number of citations or other classification criteria;
- Listing the minimum, maximum or average number of an author's co-authors according to a given publication type or year;
- Listing the number of publications where a given author is the main author, by publication type, by subject or by year;
- For a given author, knowing his/her publishing frequency by year, and knowing where he/she publishes the most (conferences, journals, books).

Currently, the DBLP database is not structured in such a way that allows data analysis. Along with this chapter, we propose a new structure for DBLP making further and richer data analysis possible. The DBLP case study raises many requirements that are worth

considering when modeling the data warehouse. Here follows the main requirements.

Complex Objects to Analyze

The objects to be analyzed may be characterized by simple linear attributes such as numerical measures or dimension attributes we find in the classical data warehouse (Kimball & Ross, 2002). However, in real life, an object may have a more complex structure (tree-like or graph-like). For instance, authors may be characterized by their names and affiliations, but publications are rather semi-structured and composed of sections, paragraphs, internal and external links, etc.

Complex Objects as Analysis Axes

Like the facts, the analysis axes may also be complex. For instance, an academic institution may be interested in evaluating authors according to their contributions, which may have a complex structure.

Objects Being Simultaneously Facts and Dimensions

In classical data warehouse modeling, facts and dimensions are treated separately. Even in symmetric models, they remain distinct at a given time. However, a need may occur to analyze an object according to objects of the same nature, and thus, one object may occur in dimension objects and facts objects simultaneously. For instance, it may be interesting to analyze the authors according to their co-authors in publications. Another example is the citation relationship between publications when we want to evaluate a publication according to its referencing publications.

Explicit and Richer Relationships Between Objects

In the classical star schema, the relationships between facts and dimensions are implicit. For instance, when relating "sales" as measures to "departments", "products" and "time" as dimensions, we know implicitly that our schema models product sales for each department during periods of time. However, in real-life applications, the relationships need to be explicit. Moreover, there may be more than one relationship between two objects. For example, we can distinguish two relationships between authors and publications: authoring and reviewing.

Complex Aggregations

Traditionally, aggregation functions such as SUM and AVERAGE deal with numerical data, but these are not the only aggregation needs we face. For example, Ravat et al (2007a) propose to aggregate documents using a function TOP_KEYWORDS that returns the most used keywords of some analyzed documents.

MAIN FOCUS

Three concepts compose the core of our model: the complex object, the relationship, and the hierarchy. In addition, we separate the definition of the (multi)dimensional model from that of the cube in one hand and the description of metadata from that of data in the other hand. In the following, we present our conceptual and logical models.

Conceptual Model

The model we define is composed of the following elements:

1. **Complex Object:** A complex object is a focus of analysis either as a subject (fact) or as an axis (dimension). An object can be unstructured, semi-structured, or structured. It can hold numerical or textual data relating to each other in different ways (linear, tree-like and graph-like).
   Formally, we refer to an object using the abbreviation "obj" and to a class of objects using "Obj". The set of object instances is noted E_{obj} where $E_{obj} = \{obj_i, i=1, n$ *where n is the number of instances of Obj}*.
2. **Relationships between objects:** A relationship is a three-tuple "R" where:
   $R = (Obj_1$, *relationship_name,* $Obj_2)$. Similarly, a relationship instance is a three-tuple "*r*" where $r = (obj_1$, *relationship_name,* $obj_2)$. The set of relationship instances is noted E_r where
   $E_r = \{r_i, i=1, m$ *where m is the number of instances of R}*.
3. **Hierarchies:** A hierarchy is an n-tuple of objects *H* where $H= (Obj_1, Obj_2, \ldots, Obj_n)$ *where*

"n" is the number of object classes composing the hierarchy. A hierarchy instance is a tuple of objects "*h*" where $h= (obj_1, obj_2, \ldots, obj_n)$ *where "n"* is the number of object classes composing the hierarchy. The set of a hierarchy instances is noted E_h where
$E_h = \{h_i, i=1, n$ *where n is the number of hierarchy instances}*.

4. **Dimensional Model for Complex Data:** we define the complex dimensional model (*CDM*) as composed of a set of complex objects, a set of relationships, and a set of hierarchies.
CDM $=(E_{Obj}, E_R, E_H)$ *where* $E_{Obj}= \{Obj_i, i =1, n$ *where "n" represents the number of objects serving as analysis subjects, analysis axes or both}*, $E_R = \{R_i, i = 1, m$ *where "m" represents the number of relationships that are interesting for analysis }*, $E_H = \{H_i, i =1, p$ *where "p" represents the number of hierarchies needed for aggregation}*. Similarly, we define a complex dimensional model instance as composed of the sets of object instances, the sets of relationship instances, and the of sets of hierarchy instances.
5. **The complex cube:** We define a complex cube as a selection among the sets E_{Obj}, E_R and E_H of the following elements:
   - One object serving as the analysis subject (fact). We note such object Obj_S
   - Many objects serving as the analysis axes (dimensions). We note respectively each object and the set of these objects Obj_A , E_{ObjA}
   - Many relationships linking Obj_S to each Obj_A. The set of these relationships is noted E_{RC}
   - Hierarchies including each Obj_A. The set of these hierarchies is noted E_{HC}.

   We define a complex cube as a four-tuple CC, where $CC = (Obj_S, E_{ObjA}, E_{RC}, E_{HC})$. Similarly, we define a complex cube instance as a four-tuple composed of:
   - The set of the instances of the object Obj_S, noted E_{objS},
   - The sets of Obj_{Ai} instances, each one noted E_{objAi},
   - The sets of relationship instances, each one noted E_{rci},
   - The sets of hierarchy instances, each one noted E_{hci}.

   $cc = (E_{objS}, \{E_{objA}i, i=1,n\}, \{E_{rcj}, j=1, m\}, \{E_{hck}, k=1, p\})$ where n is the number of axes selected for analysis, m is the number of relationships selected for analysis, and p is the number of hierarchies selected for analysis.

To illustrate the abovementioned defined concepts, we map the content of the DBLP database on our conceptual model:

Complex Objects and Classes of Objects

We identified many objects in the database falling into classes. The main classes are: "Publication", "Publication type" (article, inproceedings, incollection, …) , "Author", "Editor", "Publisher", "Proceedings", "Series", "Journal_issue", "Subject" and "Time".

Regarding the object structures, a publication of the class "Publication" for instance is identified by a "key", has a title, a modification date, page ranges, and other descriptors depending on the type of publication. Here follows an example of a publication from DBLP:

Publication 1

Key = {"conf/ISCApdcs/GrahamA03"}
Title = {Efficient Allocation in Distributed Object Databases}
Modification date = {"2006-09-20"}
Pages = {471-480}
URL = {db/conf/ISCApdcs/ISCApdcs2003.html#GrahamA03}

Relationships Between (Classes of) Objects

In DBLP, we can distinguish many relationships between the aforementioned object classes. An example of relationships with instances is:

R_1 = (Publication, "is_authored_by", Author); E_{r1} = {(publication 1, "is_authored_by", author1), (publication2, "is_authored_by", author2), …}. In this example, "author1" and "author2" refer to "Jonathan Graham" and "Jim Alves-Foss" whereas "publication 1" refers to the previous example.

Hierarchies

We can distinguish many hierarchies: publication hierarchy, conference hierarchy, journal hierarchy and time hierarchy. For instance, the conference hierarchy with instances is described as follows: H_1 = (Proceedings, Conference); E_{h1}= {(Proceedings11, conference1), (Proceedings12, conference1), (Proceedings21, conference2), …}. The corresponding objects in E_{h1} are as follows:

Proceedings11: Proceedings of Challenges, 2000 ADBIS-DASFAA Symposium on Advances in Databases and Information Systems,
Proceedings12: 5th East-European Conference on Advances in Databases and Information Systems, Vilnius, Lithuania; 25-28 September 2001
Proceedings21: ISCA PDCS 2006: San Francisco, California, USA
Conference1: Advances in Databases and Information Systems
Conference2: ISCA International Conference on Parallel and Distributed Computing Systems

Complex Dimensional Schema

Let's suppose we want to analyze the DBLP to answer the needs cited in Background section. Then, the complex dimensional schema is defined as shown in Table 1.

Complex Cube Schema

Let's now suppose we want to focus on Authors, and to analyze them in order to answer the three last questions of the needs cited in Background section. Then, the corresponding complex cube schema is defined as shown in Table 2.

Logical Model

In order to describe the logical structure of the complex dimensional model, we use XML. The main components of this logical model are described as follows:

1. **Complex Objects and classes of Objects:** We model each class of objects as an XML document. We represent an object as an XML element of a root element. The root element is named like the class but in plural whereas objects are named like the class but in singular. Each object is uniquely identified inside the document by an ID-typed XML Attribute "name" that allows referencing it from outside the document. A class document is structured as shown in Table 3.

Table 1. An example of complex dimensional schema for DBLP

CDM =(EObj, ER, EH) where EObj= {Publication, Subject, Proceedings, Journal_issue, Series, Author, Year} ER = {R1, R2, R3, R4, R5, R6} EH= {H1, H2, H3} R1 = (Publication, "Is_authored_by", Author), R2 = (Publication, "is_of_subject", Subject), R3 = (Publication, "appeared_in1", Proceedings) R4 = (Publication, "appeared_in2", Journal_issue) R5 = (Publication, "appeared_in3", Series) R6 = (Publication, "appeared_during", Year) H1 = (Publication, Publication_type) H2 = (Proceedings, Conference) H3 = (Journal_issue, Journal_volume, Journal)

Table 2. An example of a complex cube for DBLP

CC = (Obj_S, E_{ObjA}, E_{RC}, E_{HC}) Where: Obj_S= Publication E_{ObjA}= {Publication, Subject, Proceedings, Author, Year} E_{RH} = {R_1, R_2, R_3, R_6} E_{HC} = {H_1, H_2}

2. **Relationships between (classes of) objects:** We describe all the relationships in an XML document "Relationships.xml". Each relationship is then modeled as an XML element within Relationships.xml and uniquely identified by the ID-typed XML Attribute "name".
   To each relationship element in Relationships.xml corresponds an XML document named like the relationship and holding the relationship's instances. Each instance is then modeled as an XML element structured as shown in Table 4.
3. **Hierarchies:** We describe all the hierarchies in an XML document "Hierarchies.xml". Each hierarchy is then modeled as an XML element within Hierarchies.xml and uniquely identified by the ID-typed attribute "name".
   To each hierarchy in Hierarchies.xml corresponds an XML document named like the hierarchy and holding the hierarchy's instances. Each instance is then modeled as an XML element named like the hierarchy with small letters. A hierarchy instance is structured as shown in Table 5.
4. **Logical complex dimensional schema:** The complex dimensional schema description is fully based on XML which we use to define both the complex warehouse metadata and data. Thus, the warehouse metadata is composed of three XML documents: Classes.xml, Relationships.xml and Hierarchies.xml, whereas the warehouse data is disseminated through many XML documents each one named like the XML elements in the metadata documents.
5. **Logical complex cube:** We design a cube as composed of three kinds of XML documents (a) a fact document, (b) several dimension documents, and (c) several hierarchy documents.

We represent a fact object as an XML element of a root element. Furthermore, we link each fact object to its corresponding target objects using an XML element characterized by a name. This name corresponds to the ID of a relationship between the fact and the given dimension using XML sub-elements. Thus, each XML sub-element corresponds to an object linked to the fact object via the relationship evoked above. In addition, in order to avoid confusing the fact object's real data with its corresponding dimension references, we nest the former under the "Data" element and the latter under "Dimensions" element.

Table 3. The Class XML document structure

```
<class_name_in_plural>
        <class_name name = object _ID>
                …(complex structure)
        </class_name>
        …
</class_name_in_plural>
```

Table 4. The Relationship XML document structure

```
<relationship_name>
      < source target_class_ID = source_object_path/>
   < target target_class_ID = target_object_path/>
</relationship_name>
```

Table 5. The hierarchy XML document structure

```
<hierarchy_name>
        <hitem class_ID1 = object_path1 level = objec1s_level/>
        <hitem class_ID2 = object_path2 level= objec2s_level />
        …
        <hitem class_IDn = object_pathn level= objecns_level />
<hierarchy_name>
```

We represent a dimension object as an XML element. Moreover, if the dimension possesses hierarchies, we link the dimension object to its corresponding objects in each hierarchy using an XML element. This latter is characterized by a name that corresponds to the hierarchy's ID, a "*level*" XML Attribute describing the object's level in the hierarchy and sub-elements. Each sub-element corresponds to one member of the hierarchy and it is characterized by its level in the hierarchy. In addition, in order to avoid confusing the dimension object's real data with its corresponding hierarchy references, we nest the former under the "Data" element and the latter under "Hierarchies" element.

FUTURE TRENDS

Integrating complex data into the decision making process will lead to rethink many aspects of a data warehouse, from data preparation to data analysis through data management.

Moreover, using XML to describe the data warehouse at the logical level raises new challenges such as defining an XML algebra for the XML-based OLAP tools, enriching the XML query language XQuery with aggregation functions, and rethinking the performance optimization techniques of the traditional data warehouses in the context of an XML warehouse (N. Wiwatwattana, & H. V. Jagadish, Laks &V. S. Lakshmanan & Divesh Srivastava, 2007).

CONCLUSION

In this chapter, we proposed a conceptual and logical model for complex data warehouses. Our conceptual model answers new modeling requirements exemplified by the DBLP case study, which we believe are not answered by the existing models.

At the core of our model, we defined the concepts of complex object, relationship, and hierarchy. The most benefit of our model is that it gives the user plenty of possibilities to analyze the complex data while preserving its complexity in term of structure or relationships. Furthermore, the user can choose a same class of objects as an analysis subject and axis simultaneously. At the logical level, our model is fully based on XML and makes use of the main XML modeling concepts "Element" and "Attribute". At the physical level, our model offers some flexibility with respect to the physical storage. In fact, the RDF-like reference mechanism allows distributing (fragmenting) a class' instances through different documents which will have a considerable impact on query performance.

REFERENCES

Baril, X. & Bellahsène, Z. (2000). A View Model for XML Documents. *Proceedings of the 6th International Conference on Object Oriented Information Systems (OOIS 2000)*, London, UK. 429–441.

Ben Messaoud, R., Boussaïd, O., & Loudcher, S., (2006). A Data Mining-Based OLAP Aggregation of Complex Data : Application on XML Documents. *International Journal on Data warehousing and Mining, 2* (4), 1-26.

Boussaïd, O., Ben Messaoud, R., Choquet, R. & Anthoard, S., (2006). X-Warehousing: an XML-Based Approach for Warehousing Complex Data. *10th East-European Conference on Advances in Databases and Information Systems, in LNCS 4152*, 39-54. Thessaloniki, Greece.

Inmon, W.H., (2005). Building the Data Warehouse. John Wiley and Sons.

Jones, M. E., & Song, I.-Y. (2005) Dimensional modeling: identifying, classifying & applying patterns. *Proceedings of ACM 8th International Workshop on Data Warehousing and OLAP,* 29-38. Bremen, Germany.

Kimball, R., & Ross, M., (2002). The Data Warehouse Toolkit. John Wiley and Sons.

Ley, M., & Reuther, P., (2006). Maintaining an Online Bibliographical Database: The Problem of Data Quality. *Extraction et Gestion des connaissances.* 5-10. Lille, France.

Luján-Mora, S., Trujillo, J., and Song, I.-Y. (2006). A UML profile for multidimensional Modeling. *Data warehouses. Data & Knowledge Engineering, 59* (3), 725-769.

Nassis, V., Rajugan,, R., Dillon, T. S., & Rahayu, W. (2004). Conceptual Design of XML Document Warehouses. *Proceedings of the 6th International Conference Data Warehousing and Knowledge Discovery (DaWaK 2004),* Zaragoza, Spain, 1–14. Springer.

Pedersen T. B., & Jensen, C. S., (1999). Multidimensional Data Modeling for Complex Data. *Proceedings of the 15th International Conference on Data Engineering.* 336-345. Sydney, Australia.

Pokorný, J. (2001). Modelling Stars Using XML. In *Proceedings of the 4th ACM International Workshop on Data Warehousing and OLAP (DOLAP 2001)*, 24–31. Atlanta, Georgia, USA. ACM Press.

Rafanelli, M. (2003). Multidimensional Databases : Problems and Solutions. *Idea Group*.

Ravat, F., Teste, O., Tournier, R. & Zurfluh, G. (2007a). A Conceptual Model for Multidimensional Analysis of Documents. *C. Parent, K.-D. Schewe, V. C. Storey, B. Thalheim (Eds.), International Conference on Conceptual Modeling*, Springer, *LNCS 4801*, 550-565. Auckland, New Zealand.

Ravat, F., Teste, O., Tournier, R., & Zurfluh, G., (2007b). Integrating complex data into a data warehouse. *International Conference on Software Engineering and Knowledge Engineering*, 483-486. Boston, USA.

Wiwatwattana, H., Jagadish, V., Lakshmanan Laks, V. S., & Srivastava, D. (2007). "X3: A Cube Operator for XML OLAP", *23rd Intl. Conf. on Data Engineering* (*ICDE*), IEEE Computer Society, p. 916–925, 2007.

KEY TERMS

Class: Abstraction of a set of objects of the same nature.

Complex Cube Model: A class selected as the analysis' subject and surrounded by a set of classes selected as the analysis' axes where each class may have one or more related hierarchy.

Complex Dimensional Model: A set of classes related to each other with relationships and where some classes are organized as hierarchies.

Complex Object: An object characterized with a complex structure, which is interesting to the analyst either as a fact, or as an axis.

Formal Model: A model whose concepts and components are defined using formal notations amongst which there are mathematical notations.

Logical Model: A model whose concepts and components are defined using an implementation-oriented language.

Hierarchy: A set of ordered complex classes or objects extending the hierarchy concept defined between simple attributes of a dimension.

Relationship: An explicit link between complex objects or classes extending the traditional implicit link between fact and dimensions and ranging from simple associations to composition and inheritance.

ENDNOTE

[1] DBLP: http://www.informatik.uni-trier.de/~ley/db/

Multi-Group Data Classification via MILP

Fadime Üney Yüksektepe
Koç University, Turkey

Metin Türkay
Koç University, Turkey

INTRODUCTION

Data classification is a supervised learning strategy that analyzes the organization and categorization of data in distinct classes. Generally, a training set, in which all objects are already associated with known class labels, is used in classification methods. The data classification algorithms work on this set by using input attributes and builds a model to classify new objects. In other words, the algorithm predicts output attribute values. Output attribute of the developed model is categorical (Roiger & Geatz, 2003). There are many applications of data classification in finance, health care, sports, engineering and science. Data classification is an important problem that has applications in a diverse set of areas ranging from finance to bioinformatics (Chen & Han & Yu, 1996; Edelstein, 2003; Jagota, 2000). Majority data classification methods are developed for classifying data into two groups. As multi-group data classification problems are very common but not widely studied, we focus on developing a new multi-group data classification approach based on mixed-integer linear programming.

BACKGROUND

There are a broad range of methods for data classification problem including Decision Tree Induction, Bayesian Classifier, Neural Networks (NN), Support Vector Machines (SVM) and Mathematical Programming (MP) (Roiger & Geatz, 2003; Jagota, 2000; Adem & Gochet, 2006). A critical review of some of these methods is provided in this section. A major shortcoming of the neural network approach is a lack of explanation of the constructed model. The possibility of obtaining a non-convergent solution due to the wrong choice of initial weights and the possibility of resulting in a non-optimal solution due to the local minima problem are important handicaps of neural network-based methods (Roiger & Geatz, 2003). In recent years, SVM has been considered as one of the most efficient methods for two-group classification problems (Cortes & Vapnik, 1995; Vapnik, 1998). SVM method has two important drawbacks in multi-group classification problems; a combination of SVM has to be used in order to solve the multi-group classification problems and some approximation algorithms are used in order to reduce the computational time for SVM while learning the large scale of data.

There have been numerous attempts to solve classification problems using mathematical programming (Joachimsthaler & Stam, 1990). The mathematical programming approach to data classification was first introduced in early 1980's. Since then, numerous mathematical programming models have appeared in the literature (Erenguc & Koehler, 1990) and many distinct mathematical programming methods with different objective functions are developed in the literature. Most of these methods modeled data classification as linear programming (LP) problems to optimize a distance function. In addition to LP problems, mixed-integer linear programming (MILP) problems that minimize the misclassifications on the design data set are also widely studied. There have been several attempts to formulate data classification problems as MILP problems (Bajgier & Hill, 1982; Gehrlein 1986; Littschwager, 1978; Stam & Joachimsthaler, 1990). Since MILP methods suffer from computational difficulties, the efforts are mainly focused on efficient solutions for two-group supervised classification problems. Although it is possible to solve a multi-group data classification problem by solving several two-group problems, such approaches also have drawbacks including computational complexity resulting in long computational times (Tax & Duin, 2002).

MAIN FOCUS

The objective in data classification is to assign data points that are described by several attributes into a predefined number of groups. The use of hyper-boxes for defining boundaries of the sets that include all or some of the points in that set as shown in Figure 1 can be very accurate for multi-group problems. Hyper-boxes are high dimensional geometrical shapes that have lower and upper boundaries for each attribute. If it is necessary, more than one hyper-box can be used in order to represent a group as shown in Figure 1.

The data classification problem based on this idea is developed in two parts: training and testing. During the training part, characteristics of data points that belong to a certain group are determined and differentiated from the data points that belong to other groups. After the distinguishing characteristics of the groups are determined, then the effectiveness of the classification must be tested. Predictive accuracy of the developed model is performed on a test data set during the testing stage.

Training Problem Formulation

Training is performed on a training data set composed of a number of instances *i*. The data points are represented by the parameter a_{im} that denotes the value of attribute *m* for the instance *i*. The group *k* that the data point *i* belongs to are given by the set D_{ik}. Each existing hyper-box *l* encloses a number of data points belonging to group *k*. Moreover, bounds *n* (*lower, upper)* of each hyper-box is determined by solving the training problem.

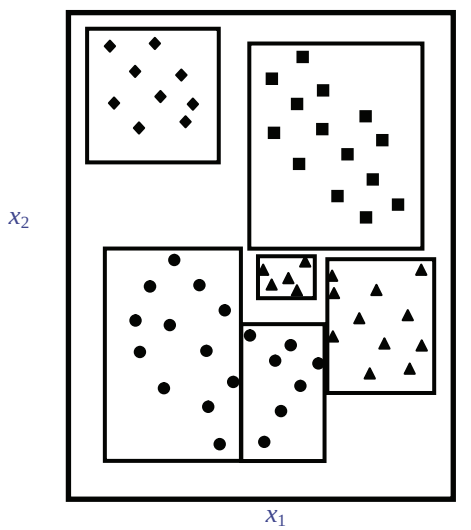
Given these parameters and the sets, the following variables are sufficient to model the multi-group data classification problem with hyper-boxes. The binary variable yb_l is indicates whether the box *l* is used or not. The position (inside or outside) of the data point *i* with regard to box *l* is represented by ypb_{il}. The assigned group *k* of box *l* and data point *i* is symbolized by ybc_{lk} and ypc_{ik}, respectively. If the data point *i* is within the bound *n* with respect to attribute *m* of box *l*, then the binary variable $ypbn_{ilmn}$ takes the value of 1, otherwise 0. Similarly, $ypbm_{ilm}$ indicates whether the data point *i* is within the bounds of attribute *m* of box *l* or not. Finally, yp_{ik} indicate the misclassification of data point *i* to group *k*. In order to define the boundaries of hyper-boxes, two continuous variables are required: X_{lmn} is the one that models bounds *n* for box *l* on attribute *m*. Correspondingly, bounds *n* for box *l* of group *k* on attribute *m* are defined with the continuous variable $XD_{l,k,m,n}$.

The following MILP problem models the training part of multi-group data classification method using hyper-boxes:

$$\min \; z = \sum_{l}\sum_{k} yp_{ik} + \sum_{l} yb_l \qquad (1)$$

subject to

Figure 1. Schematic representation of multi-group data classification using hyper-boxes.



$$XD_{lkmn} \le a_{im} ypb_{il} \quad \forall i\ k,l,m,n \,|\, n = lo \qquad (2)$$

$$XD_{lkmn} \ge a_{im} ypb_{il} \quad \forall i\ k,l,m,n \,|\, n = up \qquad (3)$$

$$XD_{lkmn} \le Q\ ybc_{lk} \quad \forall k,l,m,n \qquad (4)$$

$$\sum_k XD_{lkmn} = X_{lmn} \quad \forall l,m,n \qquad (5)$$

$$ypbn_{ilmn} \ge (1/Q)(X_{lmn} - a_{im}) \quad \forall i\ l,m,n \,|\, n = up \qquad (6)$$

$$ypbn_{ilmn} \ge (1/Q)(a_{im} - X_{lmn}) \quad \forall i\ l,m,n \,|\, n = lo \qquad (7)$$

$$\sum_l ypb_{il} = 1 \quad \forall i \qquad (8)$$

$$\sum_k ypc_{ik} = 1 \quad \forall i \qquad (9)$$

$$\sum_l ypb_{il} = \sum_k ypc_{ik} \quad \forall i \qquad (10)$$

$$\sum_k ybc_{lk} \le yb_l \quad \forall l \qquad (11)$$

$$ybc_{lk} - \sum_i ypb_{il} \le 0 \quad \forall l\ k \qquad (12)$$

$$ybc_{lk} - \sum_i ypc_{ik} \le 0 \quad \forall l\ k \qquad (13)$$

$$\sum_n ypbn_{ilmn} - ypbm_{ilm} \le \mathrm{N} - 1 \quad \forall i,l,m \qquad (14)$$

$$\sum_m ypbm_{ilm} - ypb_{il} \le \mathrm{M} - 1 \quad \forall i\ l \qquad (15)$$

$$ypc_{ik} - yp_{ik} \le 0 \quad \forall i\ k \notin D_{ik} \qquad (16)$$

$$X_{lmn}, XD_{lkmn} \ge 0,\ yb_l\ \ ybc_{lk}, ypb_{il}, ypc_{ik}, ypbn_{ilmn}, ypbm_{ilm}, yp_{ik} \in \{0,1\} \qquad (17)$$

The objective function of the MILP problem (Eq. (1)) is to minimize the misclassifications in the data set with the minimum number of hyper-boxes. The lower and upper bounds of the boxes are given in Eqs. (2) and (3), respectively. The lower and upper bounds for the hyper-boxes are determined by the data points that are enclosed within the hyper-box. Eq. (4) enforces the bounds of hyper-boxes exist if and only if this hyper-box is assigned to a group. Eq. (5) is used to relate the two continuous variables that represent the bounds of the hyper-boxes. The position of a data point with respect to the bounds on attribute *m* for a hyper-box is given in Eqs. (6) and (7). The binary variable $ypbn_{ilmn}$ helps to identify whether the data point *i* is within the hyper-box *l*. Two constraints, one for the lower bound and one for the upper bound, are needed for this purpose (Eqs. (6) and (7)). Since these constraints establish a relation between continuous and binary variables, an arbitrarily large parameter, *Q*, is included in these constraints. The Eqs. (8) and (9) state that every data point must be assigned to a single hyper-box, *l*, and a single group, *k*, respectively. The equivalence between Eqs. (8) and (9) is given in Eq. (10); indicating that if there is a data point in the group *k*, then there must be a hyper-box *l* to represent the group *k* and vice versa. The existence of a hyper-box implies the assignment of that hyper-box to a group as shown in Eq. (11). If a group is represented by a hyper-box, there must be at least one data point within that hyper-box as in Eq. (12). In the same manner, if a hyper-box represents a group, there must be at least a data point within that group as given in Eq. (13). The Eq. (14) represents the condition of a data point being within the bounds of a box in attribute *m*. If a data point is within the bounds of all attributes of a box, then it must be in the box as shown in Eq. (15). When a data point is assigned to a group that it is not a member of, a penalty applies as indicated in Eq. (16). Finally, last constraint gives non-negativity and integrality of decision variables. By using this MILP formulation, a training set can be studied and the bounds of the groups are determined for the multi-group data classification problem.

Testing Problem Formulation

The testing phase is straight forward. If a new data point whose membership to a group is not known arrives, it is necessary to assign this data point to one of the groups. There are two possibilities for a new data point when determining its group:

i. the new data point is within the boundaries of a single hyper-box,
ii. the new data point is not enclosed in any of the hyper-boxes determined in the training problem.

When the first possibility is realized for the new data point, the classification is made by directly assigning this data to the group that was represented by the

hyper-box enclosing the data point. In the case when the second possibility applies, the assignment of the new data point to a group requires further analysis. If the data point is within the lower and upper bounds of all but not one of the attributes (i.e., *m'*) defining the hyper-box, then the shortest distance between the new point and the hyper-box is calculated using the minimum distance between hyper-planes defining the hyper-box and the new data point as given in Eq. (18).

$$\min_{l,m,n}\left\{\left(a_{jm}-X_{lmn}\right)\right\} \tag{18}$$

When the data point is between the bounds of smaller than or equal to M-2 attributes, then the smallest distance between the point and the hyper-box is obtained by calculating the minimum distance between edges of the hyper-box and the new point as given in Eq. (25).

$$\vec{W}_{jlmn}=\vec{A}_j-\vec{P0}_{lmn} \tag{19}$$

$$\vec{V}_{jlmn}=\vec{P1}_{lmn}-\vec{P0}_{lmn} \tag{20}$$

$$C1_{jlmn}=(W_{jlmn}\cdot V_{jlmn})/\left\|W_{jlmn}\right\|\left\|V_{jlmn}\right\| \tag{21}$$

$$C2_{jlmn}=(V_{jlmn}\cdot V_{jlmn})/\left\|V_{jlmn}\right\|\left\|V_{jlmn}\right\| \tag{22}$$

$$b_{jlmn}=C1_{jlmn}/C2_{jlmn} \tag{23}$$

$$Pb_{jlmn}=P0_{jlmn}+b_{jlmn}V_{jlmn} \tag{24}$$

$$\min_{l,n}\left\{\sqrt{\sum_m\left(a_{jm}-pb_{jlmn}\right)^2}\right\} \tag{25}$$

When data point is not within the lower and upper bounds of any attributes defining the box, then the shortest distance between the new point and the hyper-box is calculated using the minimum distance between extreme points of the hyper-box and the new data as given in Eq. (26).

$$\min_{l,n}\left\{\sqrt{\sum_m\left(a_{jm}-X_{lmn}\right)^2}\right\} \tag{26}$$

Application

We applied the mathematical programming method on a set of 20 data points in 4 different groups given in Figure 2.

In this problem, 16 of these points were used in training and 4 of them used in testing. The training problem classified the data into four groups using 5 hyper-boxes as shown in Figure 2. It is interesting to note that Group1 requires two hyper-boxes while the other groups are represented with a single hyper-box. The reason for having two hyper-boxes for Group1 is due to the fact that a single hyper-box for this group would include one of the data points from another group, Group3. In order to eliminate inconsistencies in training data set, the method included one more box for Class1. After the training is completed successfully, the test data is processed to assign them to hyper-boxes that classify the data perfectly. The test problem also classified the data points with 100% accuracy as shown in Figure 2.

This illustrative example is also tested by different data classification models existing in the literature in order to compare the results and to measure the performance of the proposed model. Table 1 shows the examined models and their accuracies for this illustrative example. It can be concluded that the MILP approach performs better than other data classification methods that are listed in Table 1 for the illustrative example. The accuracy of the MILP approach is tested on IRIS, protein folding type and drug activity datasets. The results indicate that the MILP approach has better accuracy than other methods on these datasets [Uney & Turkay, 2005; Turkay & Uney & Yılmaz, 2006; Kahraman & Turkay, 2007; Uney & Yilmaz & Turkay, 2008].

FUTURE TRENDS

Future studies include evaluation of the performance of the proposed approach on other benchmark datasets. Moreover, by the help of artificially generated data sets, we plan to observe the performance of methods on data sets with distinct characteristics. Hence, we will determine the most suitable and unsuitable kind of data sets to this approach.

Figure 2. Hyper-boxes that classify the data points in the illustrative example.

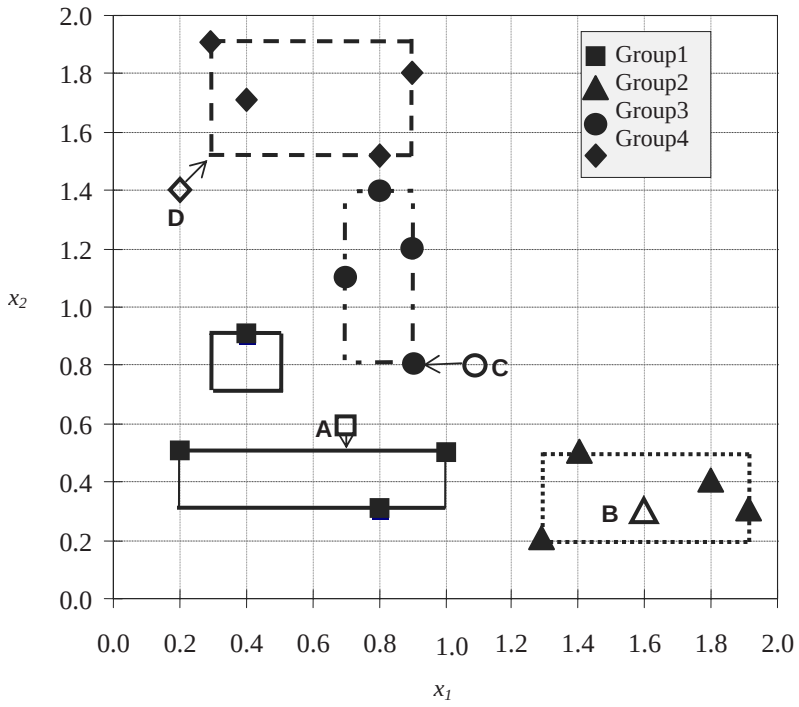


Table 1. Comparison of different classification models for the illustrative example

Classification Model	Prediction Accuracy	Misclassified Sample(s)
Neural Networks[a]	75%	A
Support Vector Machines[b]	75%	D
Bayesian Classifier[c]	75%	C
K-nearest Neighbor Classifier[c]	75%	A
Statistical Regression Classifiers[c]	75%	C
Decision Tree Classifier[c]	50%	A, C
MILP approach	100%	-

[a] iDA implementation in MS Excel [b] SVM implementation in Matlab [c] WEKA

CONCLUSION

Multi-group data classification problem can be very effectively modeled as an MILP problem. One of the most important characteristics of the MILP approach is allowing the use of hyper-boxes for defining the boundaries of the groups that enclose all or some of the points in that set. In other words, if necessary, more than one hyper-box is constructed for a specific group in the training phase. Moreover, well-construction of the boundaries of each group provides the lack of misclassifications in the training set and indirectly improves the accuracy of the model. The suggested approach has shown to be very effective on the benchmark datasets studied so far. Furthermore, the proposed model can be used for both binary and multi-group data classification problems.

REFERENCES

Adem, J. & Gochet, W. (2006). Mathematical programming based heuristics for improving LP-generated classifiers for the multi-class supervised classification problem. *European Journal of Operational Research*, 168, 181-199.

Bajgier, S. M. & Hill, A. V. (1982). An experimental comparison of statistical and linear programming approaches to the discriminant problem. *Decision Sciences*, 13, 604-618.

Chen, M. S., Han, J., & Yu, P. S. (1996). Data Mining: An overview from a database perspective. *IEEE Transactions Knowledge and Data Engineering*, 8, 866-883.

Cortes, C. & Vapnik, V. (1995). Support vector network. *Machine Learning*, 20, 273-297.

Edelstein, H. (2003). *Building Profitable Customer Relationships with Data Mining*. Two Crows Corporation.

Erenguc, S. S., Koehler, G. J. (1990). Survey of mathematical programming models and experimental results for linear discriminant analysis. *Managerial and Decision Economics*, 11, 215-225.

Gehrlein, W. V. (1986). General mathematical programming formulations for the statistical classification problem. *Operations Research Letters*, 5 (6), 299-304.

Jagota, A. (2000). *Data Analysis and Classification for Bioinformatics*. Bay Press.

Joachimsthaler, E. A. & Stam, A. (1990). Mathematical programming approaches for the .classification problem in two-group discriminant analysis. *Multivariate Behavioral Research*, 25, 427-454.

Kahraman, P. & Turkay, M. (2007). Classification of 1,4-Dihydropyridine Calcium Channel Antagonists using Hyper-Box Approach. *Ind. Eng. Chem. Res.*, 46 (14), 4921-4929.

Littschwager, J. M. & Wang, C. (1978). Integer programming solution of a classification problem. *Management Science*, 24 (14), 1515-1525.

Roiger, R. J. & Geatz, M. W. (2003). *Data Mining-A Tutorial Based Primer*. Addison Wesley Press.

Stam, A. & Joachimsthaler, E. A. (1990). A comparison of a robust mixed-integer approach to existing methods for establishing classification rules for the discriminant problem. *European Journal of Operations Research*, 46 (1), 113-122.

Tax, D. & Duin, R. (2002). Using two-class classifiers for multi class classification. *Proceedings 16th International Conference on Pattern Recognition, Quebec City, Canada, Vol. II, IEEE Computers Society Press, Los Alamitos*, 124-127.

Turkay, M., Uney, F. & Yilmaz, O. (2005). Prediction of folding type of proteins using Mixed-Integer Linear Programming, *Computer-Aided Chem. Eng., Vol 20A: ESCAPE-15, L. Puigjaner and A. Espuna (Eds.)*, 523-528, Elsevier, Amsterdam.

Uney, F. & Turkay, M. (2006). A Mixed-Integer Programming Approach to Multi-Class Data Classification Problem. *European Journal of Operational Research*, 173 (3), 910-920.

Uney Yuksektepe, F., O. Yilmaz and M. Turkay (2008). Prediction of Secondary Structures of Proteins using a Two-Stage Method, *Comput. Chem. Eng.*, 32 (1-2), 78-88.

Vapnik, V. N. (1998). *Statistical Learning Theory*. Wiley, New York.

KEY TERMS

Bayesian Classifier: Bayesian Classifier is a simple probabilistic classifier based on applying Bayes' Theorem with strong independence assumptions.

Data Classification (DC): DC is a supervised learning strategy that tries to build models which able to assign new instances to a set of well-defined classes.

Decision Tree Induction: It is a predictive model that maps observations of training instances to conclusions using a tree structure.

Linear Programming (LP): LP problems involve the optimization of a linear function subject to linear equality and/or inequality constraints.

Mathematical Programming (MP): MP refers to study of problems in which one seeks to minimize

or maximize a real function by systematically choosing the values of real or integer variables within an allowed set.

Mixed Integer Linear Programming (MILP): MILP is a special case of LP problems where some of the unknown variables are integer and some of them are continuous.

Neural Network (NN): A neural network is a data structure that attempts to simulate the behavior of neurons in a biological brain.

Support Vector Machines (SVM): SVM approach operates by finding a hyper surface that will split the classes so that the distance between the hyper surface and the nearest of the points in the groups has the largest value.

Multi-Instance Learning with MultiObjective Genetic Programming

Amelia Zafra
University of Cordoba, Spain

Sebastián Ventura
University of Cordoba, Spain

INTRODUCTION

The multiple-instance problem is a difficult machine learning problem that appears in cases where knowledge about training examples is incomplete. In this problem, the teacher labels examples that are sets (also called bags) of instances. The teacher does not label whether an individual instance in a bag is positive or negative. The learning algorithm needs to generate a classifier that will correctly classify unseen examples (i.e., bags of instances).

This learning framework is receiving growing attention in the machine learning community and since it was introduced by Dietterich, Lathrop, Lozano-Perez (1997), a wide range of tasks have been formulated as multi-instance problems. Among these tasks, we can cite content-based image retrieval (Chen, Bi, & Wang, 2006) and annotation (Qi and Han, 2007), text categorization (Andrews, Tsochantaridis, & Hofmann, 2002), web index page recommendation (Zhou, Jiang, & Li, 2005; Xue, Han, Jiang, & Zhou, 2007) and drug activity prediction (Dietterich et al., 1997; Zhou & Zhang, 2007).

In this chapter we introduce MOG3P-MI, a multiobjective grammar guided genetic programming algorithm to handle multi-instance problems. In this algorithm, based on SPEA2, individuals represent classification rules which make it possible to determine if a bag is positive or negative. The quality of each individual is evaluated according to two quality indexes: sensitivity and specificity. Both these measures have been adapted to MIL circumstances. Computational experiments show that the MOG3P-MI is a robust algorithm for classification in different domains where achieves competitive results and obtain classifiers which contain simple rules which add comprehensibility and simplicity in the knowledge discovery process, being suitable method for solving MIL problems (Zafra & Ventura, 2007).

BACKGROUND

In the middle of the 1990's, Dietterich et al. (1997) described three Axis-Parallel Rectangle (abbreviated as APR) algorithms to solve the problem of classifying aromatic molecules according to whether or not they are "musky". These methods attempted to search the appropriate axis-parallel rectangles constructed by their conjunction of features. Their best performing algorithm (iterated-discrim) started with a point in the feature space and grew a box with the goal of finding the smallest box covered at least one instance from each positive bag and no instances from any negative bag. The resulting box was then expanded (via a statistical technique) to get better results.

Following Dietterich et al.'s study, a wide variety of new methods of multi-instance learning has appeared. Auer (1997) tried to avoid some potentially hard computational problems that were required by the heuristics used in the iterated-discrim algorithm and presented a theoretical algorithm, MULTINST. With a new approach, Maron and Lozano-Perez (1998) proposed one of the most famous multi-instance learning algorithms, Diverse Density (DD), where the diverse density of a point, p, in the feature space was defined as a probabilistic measure which considered how many different positive bags had an instance near p, and how far the negative instances were from p. This algorithm was combined with the Expectation Maximization (EM) algorithm, appearing as EM-DD (Zhang & Goldman, 2001). Another study that extended the DD algorithm to maintain multilearning regression data sets was the EM-based multi-instance regression algorithm (Amar, Dooly, Goldman, & Zhang, 2001).

In 1998, Long and Tan (1998) described a polynomial-time theoretical algorithm and showed that if instances in the bags were independently drawn from product distribution, then the APR was PAC-learnable. Following with PAC-learnable research, Kalai and Blum (1998) described a reduction from the problem of PAC-learning under the MIL framework to PAC-learning with one-sided random classification noise, and presented a theoretical algorithm with less complexity than the algorithm described in Auer (1997).

The first approaches using lazy learning, decision trees and rule learning were researched during the year 2000. In the lazy learning context, Whang and Zucker (2000) proposed two variants of the k nearest-neighbour algorithm (KNN) that they referred to as Citation-KNN and Bayesian-KNN; these algorithms extended the k-nearest neighbor algorithm for MIL adopting Hausdorff distance. With respect to decision trees and learning rules, Zucker and Chevaleyre (2000) implemented ID3-MI and RIPPER-MI, which are multi-instance versions of decision tree algorithm ID3 and rule learning algorithm RIPPER, respectively. At that time, Ruffo (2000) presented a multi-instance version of the C4.5 decision tree, which was known as RELIC. Later, Zhou et al. (2005) presented the Fretcit-KNN algorithm, a variant of Citation-KNN that modified the minimal Hausdorff distance for measuring the distance between text vectors and using multiple instance perspective.
There are also many other practical multiple instance (MI) algorithms, such as the extension of standard neural networks to MIL (Zhang & Zhou, 2006). Also there are proposals about adapting Support Vector Machines to multi-instance framework (Andrews et al., 2002; Qi and Han, 2007) and the use of ensembles to learn multiple instance concepts, (Zhou & Zhang, 2007).

We can see that a variety of algorithms have been introduced to learn in multi-instance settings. Many of them are based on well-known supervised learning algorithms following works such as Ray and Craven's (2005) who empirically studied the relationship between supervised and multiple instance learning, or Zhou (2006) who showed that multi-instance learners can be derived from supervised learners by shifting their focuses from the discrimination on the instances to the discrimination on the bags. Although almost all popular machine learning algorithms have been applied to solve multiple instance problems, it is remarkable that the first proposals to adapt Evolutionary Algorithm to this scenario have not appeared until 2007 (Zafra, Ventura, Herrera-Viedma, & Romero 2007; Zafra & Ventura, 2007) even though these algorithms have been applied successfully in many problems in supervised learning.

MAIN FOCUS

Genetic Programming is becoming a paradigm of growing interest both for obtaining classification rules (Lensberg, Eilifsen, & McKee, 2006), and for other tasks related to prediction, such as characteristic selection (Davis, Charlton, Oehlschalager, & Wilson, 2006) and the generation of discriminant functions. The major considerations when applying GP to classification tasks are that a priori knowledge is not needed about the statistical distribution of the data (data distribution free). It can operate directly on the data in their original form, can detect unknown relationships that exist among data, expressing them as a mathematical expression and can discover the most important discriminating features of a class. We can find different proposals that use the GP paradigm to evolve rule sets for different classification problems, both two-class ones and multiple-class ones. Results show that GP is a mature field that can efficiently achieve low error rates in supervised learning, hence making it feasible to adapt to multiple instance learning to check its performance.

We propose, MOG3P-MI, a multiobjective grammar guided genetic programming algorithm. Our main motivations to introduce genetic programming into this field are: (a) grammar guided genetic programming (G3P) is considered a robust tool for classification in noisy and complex domains where it achieves to extract valuable information from data sets and obtain classifiers which contain simple rules which add comprehensibility and simplicity in the knowledge discovery process and (b) genetic programming with multiobjective strategy allows us to obtain a set of optimal solutions that represent a trade-off between different rule quality measurements, where no one can be considered to be better than any other with respect to all objective functions. Then, we could introduce preference information to select the solution which offers the best classification guarantee with respect to new data sets.

In this section we specify different aspects which have been taken into account in the design of the MOG3P-MI algorithm, such as individual representa-

tion, genetic operators, fitness function and evolutionary process.

Individual Representation

The choice of adequate individual representation is a very important step in the design of any evolutionary learning process to determine the degree of success in the search for solutions. In the proposed algorithm the representation is given by two components: a phenotype and a genotype. An individual phenotype represents a full classifier which is applied to bags. This classifier labels a bag as being a positive bag if it contains at least one instance which satisfies the antecedent, otherwise it is labelled as a negative bag. The representation has the structure shown in Figure 1.

The antecedent consists of tree structures and is applied to instances. It represents the individual genotype which can contain multiple comparisons attached by conjunction or disjunction according to a grammar to enforce syntactic constraints and satisfy the closure property (see Figure 2).

Genetic Operators

The elements of the following population are generated by means of two operators: mutation and crossover, designed to work in grammar guided genetic programming systems.

Mutation

The mutation operator randomly selects a node in the tree and the grammar is used to derive a new subtree which replaces the subtree in this node. If the new offspring is too large, it will be eliminated to avoid having invalid individuals. Figure 3 shows an example of this mutation.

Figure 1. Classifier applied to multi-instance learning

Cover$_{\text{bag}}$**(bag$_i$)** → *IF* $\exists$ instance$_j$ $\in$ bag$_i$ where Coverinstance(instance$_j$) is positive
THEN The bag is positive.
ELSE The bag is negative.

Coverinstance(instance$_i$) → *IF* (antecedent is satisfied by instance$_i$)
THEN The instance is positive.
ELSE The instance is negative.

Figure 2. Grammar used for individual representation

<antecedent> → <comp>
| **OR** <comp> <antecedent>
| **AND** <comp> <antecedent>

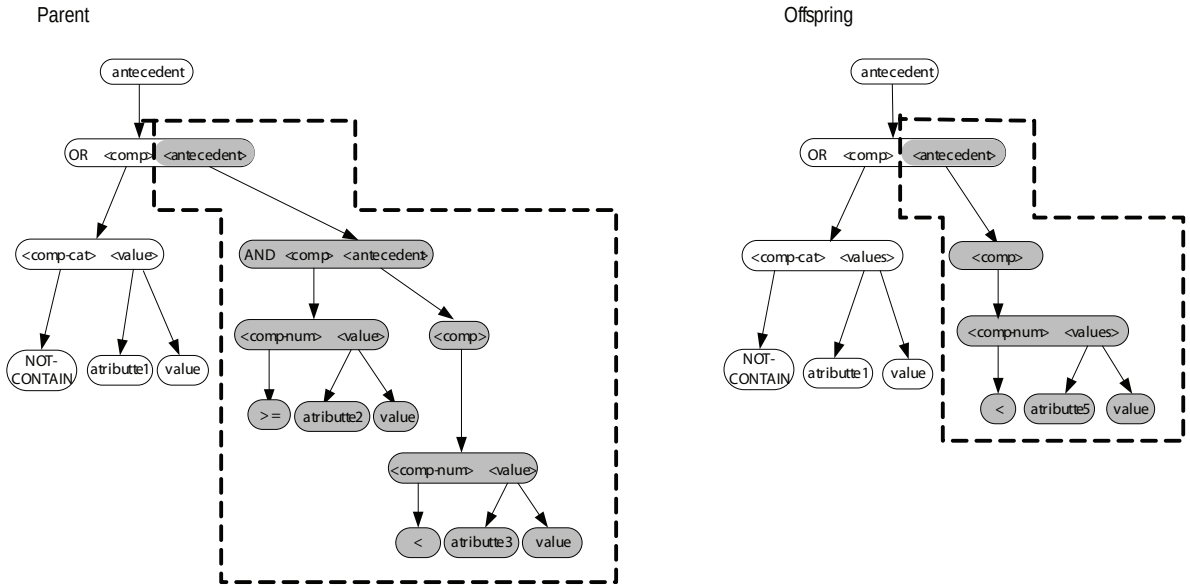
<comp> → <comp-num> <values>
| <comp-cat> <values>

<comp-num> → <
| ≥

<comp-cat > → **CONTAIN**
| **NOT_CONTAIN**

<values> → **attribute value**

Figure 3. Example of mutation process



Crossover

The crossover is performed by swapping the sub-trees of two parents for two compatible points randomly selected in each parent. Two tree nodes are compatible if their subtrees can be swapped without producing an invalid individual according to the defined grammar. If any of the two offspring is too large, they will be replaced by one of their parents. Figure 4 shows an example of the crossover operator.

Fitness Function

The fitness function evaluates the quality of each individual according to two indices that are normally used to evaluate the accuracy of algorithms in supervised classification problems. These are sensitivity and specificity. Sensitivity is the proportion of cases correctly identified as meeting a certain condition and specificity is the proportion of cases correctly identified as not meeting a certain condition.

The adaptation of these measures to the MIL field needs to consider the bag concept instead of the instance concept. In this way, their expression would be:

$$specificity = \frac{t_n}{t_n + t_p} \; ; \; sensitivity = \frac{t_p}{t_p + f_n}$$

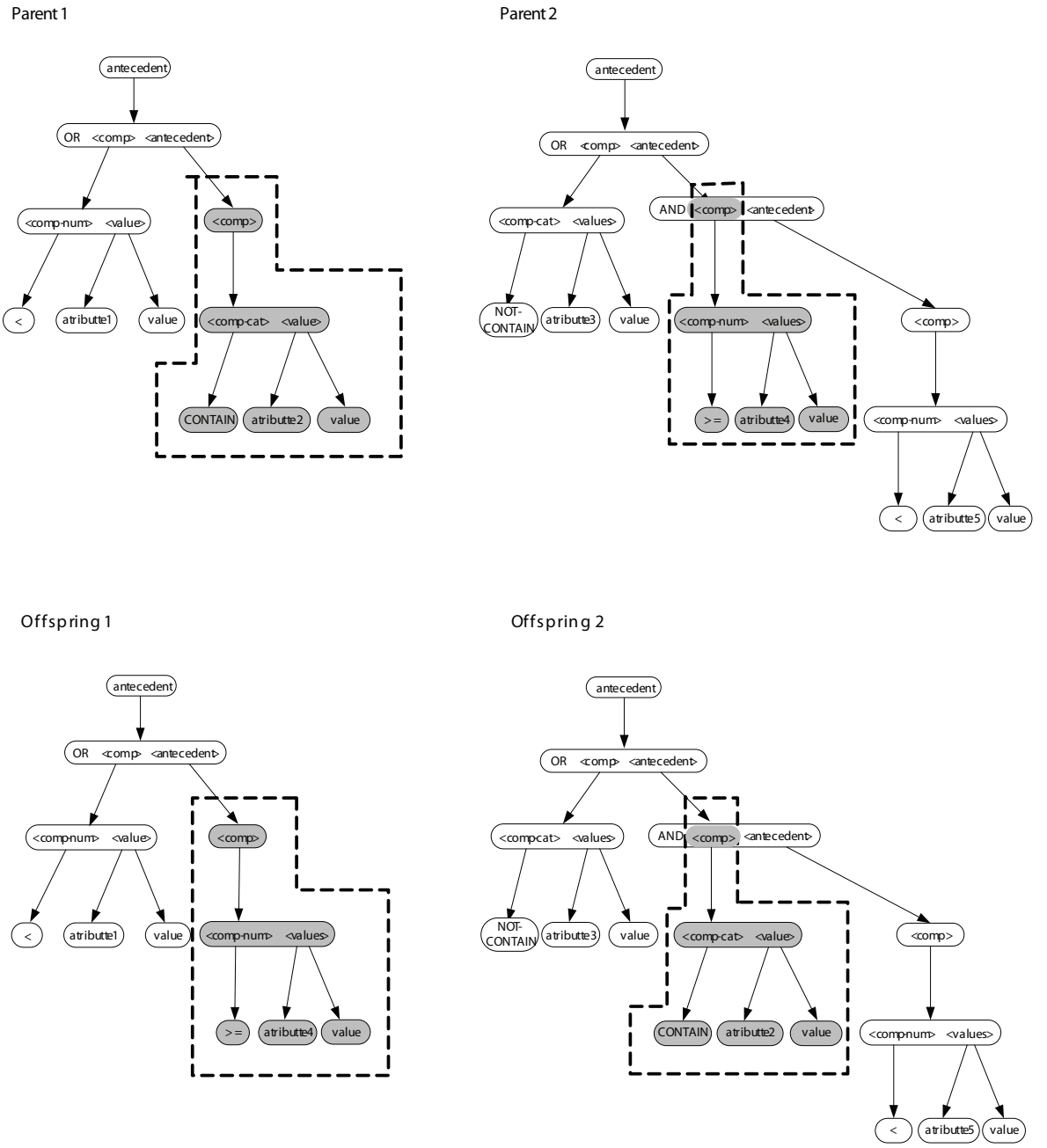
where *true positive* (t_p) represents the cases where the rule predicts that the bag has a given class and the bag does have that class. *True negative*, (t_n), are cases where the rule predicts that the bag does not have a given class, and indeed the bag does not have it. *False negative*, (f_n) cases are where the rule predicts that the bag does not have a given class but the bag does have it. Finally, P, is the number of positive bags and N, is the number of negative bags.

The evaluation involves a simultaneous optimization of these two conflicting objectives where a value of 1 in both measurements represents perfect classification. Normally, any increase in sensitivity will be accompanied by a decrease in specificity. Thus, there is no single optimal solution, and the interaction among different objectives gives rise to a set of compromised solutions, largely known as the Pareto-optimal solutions. Since none of these Pareto-optimal solutions can be identified as better than any others without further consideration, the goal is to find as many Pareto-optimal solutions as possible and include preference information to choose one of them as the final classifier.

Evolutionary Algorithm

The main steps of our algorithm are based on the well-known Strength Pareto Evolutionary Algorithm 2 (SPEA2). This algorithm was designed by Zitzler, Laumanns and Thiele (2001). It is a Pareto Front based multiobjective evolutionary algorithm that introduces some interesting concepts, such as an external elitist

Figure 4. Example of recombination process



set of non-dominated solutions, a fitness assignment schema which takes into account how many individuals each individual dominates and is dominated by, a nearest neighbour density estimation technique and a truncation method that guarantees the preservation of boundary solutions. The general outline of SPEA2 algorithm is shown in Figure 5.

FUTURE TRENDS

During these years, significant research efforts have been dedicated to MI learning and many approaches have been proposed to tackle MI problems. Although some very good results have been reported, the study of MI learning still requires topics which should be addressed. First, more datasets would have to be avail-

Figure 5. Main steps of MOG3P-MI algorithm

Algorithm MOG3P-MI

BEGIN
Generate initial population of rules, P_0 and empty archive (external set) A_0. Set t = 0.

DO
Calculate fitness values of individuals in P_t and A_t.
A_{t+1} = non-dominated individuals in P_t and A_t.
IF (size of A_{t+1} > N)
Reduce A_{t+1}.
ELSE IF (size of A_{t+1} < N)
Fill A_{t+1} ith dominated individuals in P_t and A_t.
END-IF

Fill mating pool by binary tournament selection with replacement on A_{t+1}.
Apply recombination and mutation operators to selected population, P_{t+1}.
Set P_{t+1} as resulting population. Set t = t + 1.

UNTIL an acceptable classification rule is found or the specified maximum number of generations has been reached.

END

able for the purpose of evaluation because the lack of information about many of the MI problems tackled limits studies and comparisons with other developed methods. Secondly, studies are needed to establish a general framework for MI methods and applications. Recognizing the essence and the connection between different methods can sometimes inspire new solutions with well-founded theoretical justifications. Thirdly, with respect to our adaptation of the Genetic Programming paradigm, although it has shown excellent results, more optimization of this method is possible: issues such as the stopping criterion, the pruning strategy, the choice of optimal solutions and the introduction of new objectives for further simplification would be interesting issues for future work.

CONCLUSION

The problem of MIL is a learning problem which has drawn the attention of the machine learning community. We describe a new approach to solve MIL problems which introduces the Evolutionary Algorithm in this learning. This algorithm is called MOG3P-MI and it is derived from the traditional G3P method and SPEA2 multiobjective algorithm.

MOG3P-MI generates a simple rule-based classifier that increases generalization ability and includes interpretability and simplicity in the knowledge discovered. Computational experiments (Zafra & Ventura, 2007) show that the multiobjective technique applied to G3P is an interesting algorithm for learning from multiple instance examples, finding rules which maintain a trade-off between sensitivity and specificity and obtaining the best results in terms of accuracy with respect to other existing learning algorithm.

REFERENCES

Amar, R. A., Dooly, D. R., Goldman, S. A., & Zhang, Q. (2001). Multiple-instance learning of real-valued data. *Proceedings of 18th International Conference on Machine Learning*, Massachusetts, USA, 28 June – 1 July, 3-10.

Andrews, S., Tsochantaridis, I., & Hofmann, T. (2002). Support vector machines for multiple-instance learning. *Proceedings of the 2002 Conference of the Neural Information Processing System* 15, Vancouver, Canada, 10-12 December, 561-568.

Auer, P. (1997). On Learning from Multi-Instance Examples: Empirical evaluation of a theoretical approach.

Proceedings of the 14th International Conference on Machine Learning, Nashville, Tennessee, USA, 8-12 July, 21-29.

Chen, Y., Bi, J., & Wang J. Z. (2006). MILES: Multiple-instance learning via embedded instance selection. *IEEE Transactions on Pattern Analysis and Machine Learning*, *28*(12), 1931-1947.

Davis, R. A., Charlton, A. J., Oehlschlager, S., & Wilson, J. C. (2006). Novel feature selection method for genetic programming using metabolomic 1H NMR data. *Chemometrics and Intelligent Laboratory Systems*, *81*(1), 50-59.

Dietterich, T. G., Lathrop, R. H., & Lozano-Perez, T. (1997). Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, *89*(1-2), 31-71.

Kalai, A., & Blum, A. (1998). A note on learning from multiple-instance examples. *Machine Learning*,*30*(1), 23-30.

Lensberg, T., Eilifsen, A., & McKee, T. E. (2006). Bankruptcy theory development and classification via genetic programming. *European Journal of Operational Research*, *169*(2), 677-697.

Long, P.M., & Tan, L. (1998). PAC learning axis-aligned rectangles with respect to product distributions from multiple-instance examples. *Machine Learning*, *30*(1), 7-21.

Maron, O., & Lozano-Perez, T. (1997). A framework for multiple-instance learning. *Proceedings of the 1997 Conference of the Neural Information Processing System 10*, Cambridge, MA, USA, 2-6 December, 570-576.

Qi, X., & Han, Y. (2007). Incorporating multiple svms for automatic image annotation. *Pattern Recognition*, *40*(2), 728-741.

Ray, S., & Craven, M. (2005). Supervised versus multiple instance learning: an empirical comparison. *Proceedings of the 22nd International Conference on Machine learning*, Bonn, Germany, 7-11 August, 697-704.

Ruffo, G. (2000). *Learning Single and Multiple Instance Decision Tree for Computer Security Applications*. PhD thesis, Department of Computer Science. University of Turin, Torino, Italy.

Wang, J., & Zucker, J. D. (2000). Solving the multiple-instance problem: A lazy learning approach. *Proceedings of the 17th International Conference on Machine Learning*, San Francisco, CA, USA, 29 June- 2 July, 1119-1126.

Xue, X, Han J., JianY., & Zhou .Z. (2007). Link recommendation in web index page based on multi-instance leaning techniques. *Computer Research and Development*, 44(3), 106-111.

Zafra, A., Ventura, S., Herrera-Viedma, E., & Romero, C. (2007). Multiple instance learning with genetic programming for web mining. *Proceedings of the 9th International Work-Conference on Artificial Neural Networks*, San Sebastian, Spain, 20-22 June, 919-927, *LNCS 4507*, Springer-Verlag.

Zafra, A., & Ventura, S. (2007). Multi-objective genetic programming for multiple instance learning. *Proceedings the 18th European Conference on Machine Learning*, Warsaw, Poland, 17-21 September, 790-797**,** *LNAI 4701*, Springer Verlag.

Zhang, Q., & Goldman, S. (2001). EM-DD: An improved multiple-instance learning technique. *Proceedings of the 2001 of the Conference of the Neural Information Processing System 14*, Vancouver, Canada, 3-8 December, (2001).

Zhang, M. L., & Zhou, Z. H. (2006). Adapting RBF neural networks to multi-instance learning. *Neural Processing Letters*, 23(1), 1-26.

Zhou, Z.H., Jiang, K., & Li, M. (2005). Multi-instance learning based web mining. *Applied Intelligence*, 22(2), 135-147.

Zhou, Z. H. (2006) Multi-instance learning from supervised view. *Journal Computer Science and Technology*, *21*(5), 800-809.

Zhou, Z.H., & Zhang, M.L. (2007). Solving multi-instance problems with classifier ensemble based on constructive clustering. *Knowledge and Information Systems*, *11*(2), 155-170.

Zucker, J.D., & Chevaleyre, Y. (2001). Solving multiple-instance and multiple-part learning problems with decision trees and decision rules. application to the mutagenesis problem. *Proceedings of the 14th Canadian Conference on Artificial Intelligence,* Ottawa, Canada, 7-9 June, 204-214, *LNAI 2056*, Springer-Verlag.

KEY TERMS

Evolutionary Algorithm (EA): They are search and optimization methodologies based on simulation models of natural selection, which begin with a set of potential solutions and then iteratively generate new candidates and select the fittest from this set. It has been successfully applied to numerous problems from different domains, including optimization, automatic programming, machine learning, economics, ecology, studies of evolution and learning, and social systems.

Genetic Programming (GP): An Evolutionary Algorithm that provides a flexible and complete mechanism for different tasks of learning and optimization. Its main characteristic is that it uses expression tree-based representations or functional program interpretation as its computational model.

Grammar Guided Genetic Programming (G3P): An Evolutionary Algorithm that is used for individual representation grammars and formal languages. This general approach has been shown to be effective for some natural language learning problems, and the extension of the approach to procedural information extraction is a topic of current research in the GP community.

Multi-instance Learning (MIL): It is proposed as a variation of supervised learning for problems with incomplete knowledge about labels of training examples. In MIL the labels are only assigned to *bags of instances*. In the binary case, a bag is labeled positive if *at least* one instance in that bag is positive, and the bag is labeled negative if *all* the instances in it are negative. There are no labels for individual instances. The goal of MIL is to classify unseen bags or instances based on the labeled bags as the training data.

Multiobjective Optimization Problem (MOP): The problem consists of simultaneously optimizing vector functions which maintain conflicting objectives subject to some constrained conditions.

Multiobjective Evolutionary Algorithms (MOEAs): A set of Evolutionary Algorithms suitable for solving multiobjective problems. These algorithms are well suited to multiobjective optimization problems because they are fundamentally based on biological processes which are inherently multiobjective. Multiobjective Evolutionary Algorithms are able to find optimal trade-offs in order to get a set of solutions that are optimal in an overall sense.

Strength Pareto Evolutionary Algorithm 2 (SPEA2): It is an elitist Multiobjective Evolutionary Algorithm. It is an improved version of the Strength Pareto Evolutionary Algorithm (SPEA) which incorporates a fine-grained fitness assignment strategy, a density estimation technique, and an enhanced archive truncation method. SPEA2 operates with a population (archive) of fixed size, from which promising candidates are drawn as parents of the next generation. The resulting offspring then compete with the former ones for inclusion in the population.

Supervised Learning: A machine learning technique for creating a model from training data where every training instance is assigned a discrete or real-valued label. The task of the supervised learner is to classify unseen instances based on the labelled instances of training data.

Multilingual Text Mining

Peter A. Chew
Sandia National Laboratories, USA

INTRODUCTION

The principles of text mining are fundamental to technology in everyday use. The world wide web (WWW) has in many senses driven research in text mining, and with the growth of the WWW, applications of text mining (like search engines) have by now become commonplace. In a way that was not true even less than a decade ago, it is taken for granted that the 'needle in the haystack' can quickly be found among large volumes of text. In most cases, however, users still expect search engines to return results in the same language as that of the query, perhaps the language best understood by the user, or the language in which text is most likely to be available.

The distribution of languages on the WWW does not match the distribution of languages spoken in general by the world's population. For example, while English is spoken by under 10% of the world's population (Gordon 2005), it is still predominant on the WWW, accounting for perhaps two-thirds of documents. There are variety of possible reasons for this disparity, including technological inequities between different parts of the world and the fact that the WWW had its genesis in an English-speaking country. Whatever the cause for the dominance of English, the fact that two-thirds of the WWW is in one language is, in all likelihood, a major reason that the concept of *multilingual* text mining is still relatively new. Until recently, there simply has not been a significant and widespread need for multilingual text mining.

A number of recent developments have begun to change the situation, however. Perhaps these developments can be grouped under the general rubric of 'globalization'. They include the increasing adoption, use, and popularization of the WWW in non-English-speaking societies; the trend towards political integration of diverse linguistic communities (highly evident, for example, in the European Union); and a growing interest in understanding social, technological and political developments in other parts of the world. All these developments contribute to a greater demand for multilingual text processing – essentially, methods for handling, managing, and comparing documents in multiple languages, some of which may not even be known to the end user.

BACKGROUND

A very general and widely-used model for text mining is the vector space model; for a detailed introduction, the reader should consult an information retrieval textbook such as Baeza-Yates & Ribeiro-Neto (1999). Essentially, all variants of the vector space model are based on the insight that documents (or, more generally, chunks of text) can also be thought of as vectors (or columns of a matrix) in which the rows correspond to terms that occur in those documents. The vectors/matrices can be populated by numerical values corresponding to the frequencies of occurrence of particular terms in particular documents, or, more commonly, to *weighted* frequencies. A variety of weighting schemes are employed; an overview of some of these is given in Dumais (1991). A common practice, before processing, is to eliminate rows in the vectors/matrices corresponding to 'stopwords' (Luhn, 1957) – in other words, to ignore from consideration any terms which are considered to be so common that they contribute little to discriminating between documents. At its heart, the vector space model effectively makes the assumption that the meaning of text is an aggregation of the meaning of all the words in the text, and that meaning can be represented in a multidimensional 'concept space'. Two documents which are similar in meaning will contain many of the same terms, and hence have similar vectors. Furthermore, 'similarity' can be quantified using this model; the similarity of two documents in the vector space is the cosine between the vectors for the documents. Document vectors in the vector space model can also be used for supervised predictive mining; an example is in Pang et al. (2002), where document vectors are used to classify movie reviews into 'positive' versus 'negative'.

A variant on the vector space model commonly used in text mining is Latent Semantic Analysis (LSA) (Deerwester et al., 1990). This approach takes advantage of the higher-order structure in the association of terms with documents by applying singular value decomposition (SVD) to the initial term-by-document matrix. SVD is a method in multilinear algebra for decomposition of a matrix into its principal components, and is used to find the best lower-rank approximation to the original matrix. In text mining, SVD can be used to '[represent] both terms and documents as vectors in a space of choosable dimensionality' (Deerwester et al., 1990: 395). The principal contribution of LSA is that it deals with the inherently noisy characteristics of text by mapping terms to more nebulous 'concepts' without abandoning statistical principles to do so; a thesaurus is not required to find terms of similar meaning, as terms of similar meaning should in theory be identifiable by the statistics of their distributions. When applied to practical text mining problems, the results of LSA analyses have often been shown to be as good as, or better than, simpler vector space approaches (Deerwester et al., 1990). Perhaps the most significant advantage of LSA, however, is that the representation of documents is generally much more economical than under approaches in which each term corresponds to the 'dimension' of a vector. In typical real-world text mining problems, document sets may contain thousands of distinct terms, meaning that, with simple vector-space approaches, document vectors must contain thousands of entries (although the vectors are usually sparse). With LSA, on the other hand, documents are often represented by vectors containing on the order of 200-300 values. When it comes to computing cosines between vectors or training predictive data mining models, for example, this can save a significant amount of computational horsepower.

MAIN FOCUS

The basic problem of multilingual text mining (also known as cross-language information retrieval) is that of representing text from different languages in a single, coherent conceptual space. If this can be achieved, then, at least in theory, the groundwork is laid for solving problems such as the following:

- Computation of the similarity of documents in different languages
- Prediction of other variables (such as positive or negative sentiment) from the text, regardless of the text's language
- Clustering of documents in multiple languages by topic, regardless of the documents' languages

From a multilingual perspective, the vector space model in general has many practical features which recommend it. First, the process of tokenization – computing which terms occur in which documents – is very general, from a linguistic point of view. While linguistically the concept of the 'word' may carry different connotations from language to language, it is computationally straightforward to define the 'term' (which at least in English often corresponds to the word) in a way which can easily be applied to virtually all languages. The task is made easier by the fact that a computational infrastructure for this already exists: both Unicode, itself a product of the 'globalization' of computing, and regular expressions, facilitate tokenization in multiple languages. Essentially, we can say that a term is any portion of text bounded at its left and right edges by 'non-word' characters (the latter being pre-defined in the regular expressions framework for virtually all Unicode code pages). Thus, Unicode allows this definition to be applied equally for languages in Roman and non-Roman script; for languages with left-to-right or right-to-left script; and so on. Of the major languages of the WWW, Chinese is perhaps the only one that presents a challenge to the general definition, as 'non-word' characters often occur only at sentence boundaries in Chinese. Yet this is by no means an insurmountable problem; while Chinese characters do not exactly correspond to English words, a reasonable approach with Chinese, representing only a minor deviation from the general rule proposed above, would be to treat each *character* as a term, since Chinese characters in general each have a meaning of their own.

The simplest vector space model, in which the rows of the vectors and matrices correspond one-to-one to distinct terms, is inappropriate for multilingual text mining, however. The reason for this is that there is usually comparatively little overlap between languages with respect to the terms that occur in those languages. Even where it exists, some of the overlap is attributable to 'faux amis' – words which are homographic across languages but have different meanings, such as

English ‘coin’ and French ‘coin’ (corner). For pairs of languages written in mutually exclusive scripts, such as English and Russian, the respective sets of terms are completely disjoint by definition. Thus, an approach to cross-language text comparison which treats terms as features is more or less doomed from the start.

A more promising vector-space approach to multilingual text mining is offered by LSA. In this approach, described in Berry et al. (1994), Young (1994), and Dumais et al. (1997), the initial term-by-document matrix is formed from a parallel corpus. The list of indexed terms, then, will contain all distinct terms from *any* of the parallel languages, while each ‘document’ will be a parallel text chunk consisting of the *concatenation* of the text in all the parallel languages. The effect of this is that terms which are synonymous but from different languages (translations of one another) will tend to co-occur in the same documents. The SVD is computed for the term-by-document matrix, just as if the matrix had been formed from documents of just one language. Since SVD takes into account the statistics of term co-occurrence, terms and documents which are translations of one another will generally be represented within the vector space by vectors with a high similarity to one another. Under this method, even documents which are not in a parallel corpus can be compared to one another. Moreover, the approach is extensible: it can be applied to large numbers of languages at once, not just pairs of languages, as long as all languages are represented in the parallel corpus used to train LSA. Using parallel translations of the Bible (Resnik, 1999; Biola University, 2007) as training data, Chew & Abdelali (2007) show that the method can be extended to dozens of languages at once. Including large numbers of languages in the training data not only allows documents in all those languages to be mapped within a single cross-language semantic space; it also results in an improved characterization of that semantic space, since including more parallel languages measurably increases precision in cross-language retrieval. Increasingly, as parallel text becomes more available on-line, new possibilities are opened up for this type of analysis.

However, even multilingual LSA suffers from certain drawbacks. Essentially, these all stem from the fact that there is no simple set of correspondences between terms in one language and terms in another. For example, English ‘in’ might sometimes be translated into French as ‘en’, sometimes as ‘dans’, sometimes (together with following ‘the’) as ‘au’ or ‘aux’, and so on. Conversely, French ‘aux’ can be translated back into English as ‘in the’, ‘at the’, or ‘to the’, depending on the context. Furthermore, as can be seen from these examples, one term in French may correspond to two terms in English, meaning the word-alignment is not always one-to-one. Depending on how closely related two languages are, these effects may be more extreme. For example, Russian, unlike English, preserves a nominal case system in which the ending of a noun is determined by its function in the sentence. Some examples are given in Table 1.

In Russian, therefore, the word-endings take on many of the same functions that are fulfilled in English by prepositions. Within a model in which terms are strings of text bounded by non-word characters, the consequence of this is that Russian packs more information into each term than English. A further consequence is that Russian requires fewer terms overall to express the same meaning as English, which can be confirmed by inspection of any parallel English-Russian text. The problem of lack of simple term-to-term correspondences cannot simply be solved by removing stopwords (such as English ‘to’, ‘of’ and ‘the’ in the above example), because one must still deal with the fact that ‘собака’, ‘собаку’, ‘собаки’ and ‘собаке’ are distinct terms all corresponding to ‘dog’.

For LSA, one reason this presents a problem is that languages will not receive equal weighting in the decomposition of parallel text. If, for example, parallel English-Russian text consistently contains more English terms than Russian ones, simply because Russian requires fewer terms to convey the same meaning,

Table 1. Example term correspondences between English and Russian

‘dog’ (subject of sentence)	собака
‘dog’ (object of sentence)	собаку
‘of the dog’	собаки
‘to the dog’	собаке

English terms will tend to be overweighted in the decomposition.

An alternative to LSA which goes some way towards remedying this is PARAFAC2 (Harshman, 1972). In multilingual text mining, PARAFAC2 is a close cousin to LSA, and is also within the general family of techniques that operate within the vector-space model. PARAFAC2 assumes that the parallel corpus is represented as a three-way array instead of as a single term-by-document matrix (Chew et al., 2007). Each slice of the array corresponds to the term-by-document matrix for a single language. A separate LSA-like decomposition is performed for each slice with the constraint that the 'document-by-concept matrix' output by the decomposition (the so-called V matrix) is constant across all languages. This allows each language some autonomy in contributing to the overall decomposition, and providing a principled statistical method allowing for the fact that some languages use more terms than others. It has been demonstrated empirically that PARAFAC2 results in an improved characterization of the documents within a semantic space, compared to LSA (Chew et al., 2007); this is measured by increased precision in the results of cross-language information retrieval tests from, for example, 0.268 to 0.415. Nonetheless, the results do still vary depending on the language pair; generally, for morphologically complex languages like Russian where terms are more content-rich, results are not as good. In this respect, PARAFAC2 still has some of the same weaknesses as LSA.

FUTURE TRENDS

As the volume and diversity of on-line multilingual content increases, multilingual text mining can be expected to receive growing attention from users and the research community. To some extent, this is already becoming evident through the existence of specialist groups such as the Cross-Language Evaluation Forum (CLEF) (see for example Peters, 2001), which has explored issues surrounding cross-language information retrieval since the turn of the millennium. Interest in multilingual text mining can also be expected to grow as the processes of globalization make users increasingly aware of text resources available elsewhere around the world.

It is frequently observed that the lack of more multilingual resources (such as parallel corpora, particularly for minority languages) is an impediment to multilingual text mining. It is certainly true that recent successes in multilingual natural language processing have come from data-driven approaches such as Statistical Machine Translation (SMT) (Koehn et al. 2003, Chiang 2005). Indeed, the whole field of machine translation has undergone a revival in recent years precisely because training data is becoming more available; an additional factor is the increased availability of computing power necessary to train SMT or similar algorithms on large amounts of data. Furthermore, it is generally the case that more training data equates to improved results for methods such as SMT. Thus, to the extent that multilingual corpora are *not* available, it would be true that multilingual text mining has been held back. However, as observed in this article, and evidenced by an increasing number of workshops devoted to the issue of multilingual resources, examples being the Workshop on Multilingual Language Resources and Interoperability (Association for Computational Linguistics, 2006) and the Workshop on Cross-Lingual Information Access to be held in 2008, the problem of lack of data is in the process of being addressed. If the availability of data is not already a solved problem, it soon may be; and attention will increasingly need to be focused on the next issue at hand, which is the improvement of algorithms for processing that data.

CONCLUSION

This article has surveyed the state of the art in multilingual text mining, from an introduction to the vector space model, to techniques such as Latent Semantic Analysis and PARAFAC2. While significant strides have been made in the field, by no means all the problems of multilingual text mining have been solved. In particular, a challenge is to find a model which is more able to handle text in morphologically complex languages than existing models. One of the factors which has contributed to progress in other related fields, such as machine translation, is an expansion of textual data available for training statistical models. There is good reason to hope that these same factors, together with improved algorithms for analysis of (or decomposition of) the data will lead to new breakthroughs in the field of multilingual text mining.

ACKNOWLEDGMENT

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under Contract DE-AC04-94AL85000.

REFERENCES

Association for Computational Linguistics. (2006). *Proceedings of the Workshop on Multilingual Language Resources and Interoperability*. Association for Computational Linguistics: Stroudsburg, PA, 2006.

Baeza-Yates, R. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. New York: ACM Press.

Berry, M. W., Dumais, S. T., and O'Brien, G. W. (1994). Using Linear Algebra for Intelligent Information Retrieval. *SIAM: Review* 37, 573-595.

Biola University. (2007). *The Unbound Bible*, 2005-2006. Accessed at http://www.unboundbible.com/ on February 27, 2007.

Chew, P.A. and Abdelali, A. (2007). Benefits of the 'Massively Parallel Rosetta Stone': Cross-Language Information Retrieval with over 30 Languages. *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, 872-879.

Chew, P.A., Kolda, T. G., Bader, B. W. and Abdelali, A. (2007). Cross-Language Information Retrieval Using PARAFAC2. *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 143-152.

Chiang, D. (2005) A Hierarchical Phrase-Based Model for Statistical Machine Translation. *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, 263-270.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science* 41:6, 391-407.

Dumais, S. T. (1991). Improving the Retrieval of Information from External Sources. *Behavior Research Methods, Instruments, and Computers* 23:2, 229-236.

Dumais, S. T., Letsche, T. A., Littman, M. L., and Landauer, T. K. (1997). Automatic Cross-Language Retrieval Using Latent Semantic Indexing. *AAAI Spring Symposium on Cross-Language Text and Speech Retrieval.*

Gordon, Raymond G., Jr. (ed.). (2005). *Ethnologue: Languages of the World, 15th Ed*. Dallas, TX: SIL International. Online version: http://www.ethnologue.com/.

Harshman, R. A. (1972). PARAFAC2: Mathematical and Technical Notes. *UCLA Working Papers in Phonetics* 22, 30-47.

Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical Phrase-Based Translation. *Proceedings of the Joint Conference on Human Language Technologies and the Annual Meeting of the North American Chapter of the Association of Computational Linguistics (HLT/NAACL)*, 48-54.

Luhn, H. P. (1957). The Automatic Creation of Literature Abstracts. *IBM Journal of Research and Development* 2:2 (159-165; 317).

Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs Up? Sentiment Classification using Machine Learning Techniques. *Proceedings of EMNLP 2002*, 79-86.

Peters, C. (ed.) (2001). Cross-Language Information Retrieval and Evaluation. *Workshop of the Cross-Language Evaluation Forum, CLEF 2000*. Berlin: Springer-Verlag.

Resnik, P., Olsen, M. B., and Diab, M. (1999). The Bible as a Parallel Corpus: Annotating the "Book of 2000 Tongues". *Computers and the Humanities* 33, 129-153.

Young, P. G. (1994). *Cross Language Information Retrieval Using Latent Semantic Indexing*. Master's thesis, University of Knoxville: Knoxville, TN, 1994.

KEY TERMS

Information Retrieval: The science of searching for information in documents or for documents themselves, for example through efficient indexing of the documents.

Latent Semantic Analysis (LSA): Also known as Latent Semantic Indexing, LSA is a statistically-based natural language processing technique for analyzing relationships between documents and their constituent terms by computing the *n* most important 'concepts' expressed in the text (where *n* is determined by the analyst).

PARAFAC2: An abbreviation for Parallel Factor Analysis 2, PARAFAC2 is a model for decomposition of multi-way arrays. It is a variant of PARAFAC1 (or simply PARAFAC) which can be used for decomposition of tensors, with the difference that, under PARAFAC2, strict trilinearity is not required.

Stopwords: Terms filtered out prior to processing of text. Typically, stopwords are hand-picked according to the judgement of the analyst; they are the terms considered to have low intrinsic meaning because of their widespread distribution.

Terms: The units of which text is composed. Often, 'term' is synonymous with 'word'.

Text Mining: The automated extraction of useful information from unstructured text data.

Tokenization: The process of breaking a document or chunk of text into its constituent terms.

Vector Space Model: An algebraic approach to the representation of documents or text chunks. Documents are represented as vectors of identifiers, such as index terms.

Multiple Criteria Optimization in Data Mining

Gang Kou
University of Electronic Science and Technology of China, China

Yi Peng
University of Electronic Science and Technology of China, China

Yong Shi
CAS Research Center on Fictitious Economy and Data Sciences, China & University of Nebraska at Omaha, USA

INTRODUCTION

Multiple criteria optimization seeks to simultaneously optimize two or more objective functions under a set of constraints. It has a great variety of applications, ranging from financial management, energy planning, sustainable development, to aircraft design. Data mining is aim at extracting hidden and useful knowledge from large databases. Major contributors of data mining include machine learning, statistics, pattern recognition, algorithms, and database technology (Fayyad, Piatetsky-Shapiro, & Smyth, 1996). In recent years, the multiple criteria optimization research community has actively involved in the field of data mining (See, for example: Yu 1985; Bhattacharyya 2000; Francisci & Collard, 2003; Kou, Liu, Peng, Shi, Wise, & Xu, 2003; Freitas 2004; Shi, Peng, Kou, & Chen, 2005; Kou, Peng, Shi, Wise, & Xu, 2005; Kou, Peng, Shi, & Chen, 2006; Shi, Peng, Kou, & Chen, 2007).

Many data mining tasks, such as classification, prediction, clustering, and model selection, can be formulated as multi-criteria optimization problems. Depending upon the nature of problems and the characteristics of datasets, different multi-criteria models can be built. Utilizing methodologies and approaches from mathematical programming, multiple criteria optimization is able to provide effective solutions to large-scale data mining problems. An additional advantage of multi-criteria programming is that it assumes no deterministic relationships between variables (Hand & Henley, 1997).

BACKGROUND

The goal of data mining is to identify hidden, interesting, and useful structures from large databases (Fayyad & Uthurusamy, 2002). Methods and techniques from multiple disciplines, such as machine learning, statistics, pattern recognition, and database technology, have been applied extensively in data mining to extract patterns from data. Recently, the multi-criteria or multi-objective optimization-based methods have been proposed as another option for data mining tasks. For instance, Bhattacharyya proposed a multi-objective model for direct marketing (2000); Francisci and Collard addressed the interestingness measure of dependency rules by formulating the scenario as a multi-criteria problem (2003); and Kou, Peng, Shi, and Chen built a multi-criteria convex quadratic programming model for credit portfolio management (2006).

If a data mining task can be modeled as optimization problems with multiple objective functions, it can be cast into the multi-criteria optimization framework. Many data mining functionalities, such as classification, prediction, and interestingness measure, can be formulated as multi-criteria optimization problems. For example, in multi-criteria optimization context, the classification problem can be stated as one of simultaneously minimizing misclassified points and maximizing correctly classified points. The established methodologies and procedures for solving multi-criteria optimization problems and incorporating the results into the business decision process by the discipline of multi-criteria decision making (MCDM) can be applied to these data mining tasks.

MAIN FOCUS

Currently, the main focuses of multiple criteria optimization in data mining include: model construction, algorithm design, and results interpretation and application.

Model Construction

Model construction refers to the process of establishing mathematical models for multi-criteria data mining problems, which exist in many data mining tasks. For example, in network intrusion detection, the goal is to build classifiers that can achieve not only high classification accuracy, but also low false alarm rate. Although multiple objectives can be modeled separately, they normally can not provide optimal solutions to the overall problem (Fonseca & Fleming, 1995). Furthermore, a model may perform well on one objective, but poorly on other objectives. In this kind of scenario, multiple criteria optimization can be used to build models that can optimize two or more objectives simultaneously and find solutions to satisfy users' preferences.

Algorithm Design

Algorithm design is a set of steps that takes raw data as input and generates solutions as output. Specifically, algorithm design normally includes data preparation, optimization approach, and model assessment.

1. **Data preparation:** Raw data are selected and cleaned according to the requirements of data mining tasks. In addition, data need to be formatted into appropriate forms. Since multiple criteria optimization models can handle only numeric inputs, categorical attributes need to be transformed into numeric types.
2. **Optimization approach:** There are three main approaches to multiple criteria optimization (Freitas, 2004): (i) convert multiple objectives into a single-criterion problem using weight vectors; (ii) prioritize objectives and concentrate on objectives with high priorities, which is called the lexicographical approach; (iii) find a set of non-dominated solutions and allow business users to pick their desired solutions, which is also known as the Pareto approach. Each approach has its advantages and disadvantages. The first approach, reformatting multi-criteria as a single-objective problem, is by far the most popular one in data mining field due to its simplicity and efficiency.
3. **Model assessment:** Results of models, such as accuracy and generality, are assessed according to predefined criteria.

Results Interpretation and Application

Depending on application domains and user preferences, results should be interpreted differently. Take health insurance fraud detection as an example. Multiple criteria optimization can provide class labels and probability scores. Health insurance companies need to process large volumes of records and have limited resources to manually investigate potential fraud records (Peng et al., 2007). In this case, class labels alone, which distinguish normal records from fraud records, are not as useful as the combination of class labels and probability scores that can rank potential fraud records. Data miners should discuss with business users to determine which forms of results can better satisfy business objectives (Chapman, Clinton, Khabaza, Reinartz, & Wirth, 1999).

FUTURE TRENDS

The application of multiple criteria optimization techniques can be expanded to more data mining tasks. So far classification task is the most studied problem in the data mining literature. Multiple criteria optimization could be applied to many other data mining issues, such as data preprocessing, clustering, model selection, and outlier detection (Mangasarian, 1996). Another direction is to examine the applicability of the lexicographical and Pareto approaches in large scale data mining problems.

CONCLUSION

Many data mining tasks involve simultaneously satisfy two or more objectives and thus multiple criteria optimization is appropriate for these tasks in nature. The main components of multiple criteria optimization in data mining include model construction, algorithm design, and results interpretation and application. Model construction builds multi-criteria models; algorithm

design takes raw data and generates optimization solutions; optimization solutions are interpreted and implemented according to end users' preferences. Compare with popular data mining techniques, such as machines learning and statistics, multiple criteria optimization is not well studied. Much more work is needed to broaden the applicability of multiple criteria optimization in important data mining problems.

REFERENCES

Bhattacharyya, S. (2000). Evolutionary algorithms in data mining: multi-objective performance modeling for direct marketing. *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 465-473). New York: ACM Press.

Chapman, P., Clinton, J., Khabaza,T., Reinartz,T., & Wirth, R. (1999). The CRISP-DM Process Model, www.crisp-dm.org/Process/index.htm.

Fayyad, U. M., Piatetsky-Shapiro, G., and Smyth, P. (1996). From Data Mining to Knowledge Discovery: An Overview. In Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R. (eds.), *Advances in Knowledge Discovery and Data Mining* (pp. 1-34), Menlo Park: AAAI Press.

Fayyad, U. & Uthurusamy, R. (2002). Evolving data mining into solutions for insights. *Communications of The ACM*, 45(8), 28-31.

Fonseca, C. M. & Fleming, P. J. (1995). An Overview of Evolutionary Algorithms in Multi-Objective Optimization. *Evolutionary Computation*, 3 (1), 1-16.

Francisci, D. & Collard, M. (2003). Multi-Criteria Evaluation of Interesting Dependencies according to a Data Mining Approach. *Proceedings of the 2003 Congress on Evolutionary Computation (CEC'2003) Vol. 3* (pp. 1568-1574). Canberra, Australia: IEEE Press.

Freitas, A. A. (2004). A critical review of multi-objective optimization in data mining: a position paper. *ACM SIGKDD Explorations Newsletter*, 6(2), 77-86.

Hand, D. J., & Henley, W. E. (1997). Statistical classification methods in consumer credit scoring: a review. *Journal of the Royal Statistical Society, Series A (Statistics in Society)*, 160(3), 523-541.

International Society on Multiple Criteria Decision Making, available online at: http://project.hkkk.fi/MCDM/intro.html

Kou, G., Liu, X., Peng, Y., Shi, Y., Wise, M. and Xu, W. (2003). Multiple Criteria Linear Programming to Data Mining: Models, Algorithm Designs and Software Developments, Optimization Methods and Software 18 (4): 453-473, Part 2.

Kou, G., Peng, Y., Shi, Y., Wise, M., and Xu, W. (2005). Discovering Credit Cardholders' Behavior by Multiple Criteria Linear Programming. *Annals of Operations Research* 135 (1), 261-274.

Kou, G., Peng, Y., Shi, Y., & Chen, Z. (2006). A New Multi-criteria Convex Quadratic Programming Model for Credit Analysis. In V. N. Alexandrov et al. (Ed.), *LNCS 3994* (pp. 476 – 484). New York: Springer.

Mangasarian, O. L. (1996). *Mathematical programming in data mining* (Tech. Rep. 96-05). Madison, University of Wisconsin, Computer Science Department.

Peng, Y., Shi, Y., & Xu, W. (2002). Classification for Three-group of Credit Cardholders' Behavior via A Multiple Criteria Approach. *Advanced Modeling and Optimization* (Online Journal: http://www.ici.ro/camo/journal/v4n1.htm), Vol. 4, 39-56.

Peng, Y., Kou, G., Sabatka, A., Matza, J., Chen, Z., Khazanchi, D. & Shi, Y. (2007). Application of Classification Methods to Individual Disability Income Insurance Fraud Detection. In Y. Shi et al. (Eds.): ICCS 2007, Part III, LNCS 4489 (pp. 852 – 858), Springer-Verlag Berlin Heidelberg.

Peng, Y., Kou, G., Shi, Y., and Chen, Z. (2008), A Multi-criteria Convex Quadratic Programming model for Credit Data Analysis, *Decision Support Systems* (To appear)

Shi, Y., Peng, Y., Kou, G., & Chen, Z. (2005). Classifying Credit Card Accounts for Business Intelligence and Decision Making: a Multiple-criteria Quadratic Programming Approach. *International Journal of Information Technology and Decision Making*, *4*(4), 581-600.

Shi, Y., Peng, Y., Kou, G., & Chen, Z. (2007). Introduction to Data Mining Techniques via Multiple Criteria Optimization Approaches and Applications. In D. Taniar (Eds.), *Advanced Topics in Data Warehousing and Mining* (pp242-275). Idea Group Publishing.

Yu, P., (1985) *Multiple Criteria Decision-Making: Concepts, Techniques and Extensions*, Plenum, New York, New York, November (388 pages).

KEY TERMS

Multi-Criteria Decision Making: the study of methods and procedures by which concerns about multiple conflicting criteria can be formally incorporated into the management planning process (International Society on MCDM).

Multi-Criteria Optimization: In mathematics, optimization/mathematical programming is concerned with the problems/processes in finding the maxima and minima of objectives/functions, subject to constraints/sets. Linear programming (LP) and nonlinear programming (NLP) are two sub-areas of optimization.

Linear Programming: Linear programming (LP) problems are optimization problems with linear objective function and linear constraints.

Nonlinear Programming: Nonlinear programming problems are optimization problems with nonlinear function in the objective function, constraints or both of them.

Multiple Hypothesis Testing for Data Mining

Sach Mukherjee
University of Oxford, UK

INTRODUCTION

A number of important problems in data mining can be usefully addressed within the framework of statistical hypothesis testing. However, while the conventional treatment of statistical significance deals with error probabilities at the level of a single variable, practical data mining tasks tend to involve thousands, if not millions, of variables. This Chapter looks at some of the issues that arise in the application of hypothesis tests to multi-variable data mining problems, and describes two computationally efficient procedures by which these issues can be addressed.

BACKGROUND

Many problems in commercial and scientific data mining involve selecting objects of interest from large datasets on the basis of numerical relevance scores ("object selection"). This Section looks briefly at the role played by hypothesis tests in problems of this kind. We start by examining the relationship between relevance scores, statistical errors and the testing of hypotheses in the context of two illustrative data mining tasks. Readers familiar with conventional hypothesis testing may wish to progress directly to the main part of the Chapter.

As a topical example, consider the differential analysis of gene microarray data (Piatetsky-Shapiro & Tamayo, 2004; Cui & Churchill, 2003). The data consist of expression levels (roughly speaking, levels of activity) for each of thousands of genes across two or more conditions (such as healthy and diseased). The data mining task is to find a set of genes which are differentially expressed between the conditions, and therefore likely to be relevant to the disease or biological process under investigation. A suitably defined mathematical function (the t-statistic is a canonical choice) is used to assign a "relevance score" to each gene and a subset of genes selected on the basis of the scores. Here, the objects being selected are genes.

As a second example, consider the mining of sales records. The aim might be, for instance, to focus marketing efforts on a subset of customers, based on some property of their buying behavior. A suitably defined function would be used to score each customer by relevance, on the basis of his or her records. A set of customers with high relevance scores would then be selected as targets for marketing activity. In this example, the objects are customers.

Clearly, both tasks are similar; each can be thought of as comprising the assignment of a suitably defined relevance score to each object and the subsequent selection of a set of objects on the basis of the scores. The selection of objects thus requires the imposition of a threshold or cut-off on the relevance score, such that objects scoring higher than the threshold are returned as relevant. Consider the microarray example described above. Suppose the function used to rank genes is simply the difference between mean expression levels in the two classes. Then the question of setting a threshold amounts to asking how large a difference is sufficient to consider a gene relevant. Suppose we decide that a difference in means exceeding x is 'large enough': we would then consider each gene in turn, and select it as "relevant" if its relevance score equals or exceeds x. Now, an important point is that the data are random variables, so that if measurements were collected again from the same biological system, the actual values obtained for each gene might differ from those in the particular dataset being analyzed. As a consequence of this variability, there will be a real possibility of obtaining scores in excess of x from genes which are in fact *not* relevant.

In general terms, high scores which are simply due to chance (rather than the underlying relevance of the object) lead to the selection of irrelevant objects; errors of this kind are called false positives (or Type I errors). Conversely, a truly relevant object may have an unusually low score, leading to its omission from the final set of results. Errors of this kind are called false negatives (or Type II errors). Both types of error are associated with identifiable costs: false positives lead to wasted

resources, and false negatives to missed opportunities. For example, in the market research context, false positives may lead to marketing material being targeted at the wrong customers; false negatives may lead to the omission of the "right" customers from the marketing campaign. Clearly, the rates of each kind of error are related to the threshold imposed on the relevance score: an excessively strict threshold will minimize false positives but produce many false negatives, while an overly lenient threshold will have the opposite effect. Setting an *appropriate* threshold is therefore vital to controlling errors and associated costs.

Statistical hypothesis testing can be thought of as a framework within which the setting of thresholds can be addressed in a principled manner. The basic idea is to specify an acceptable false positive rate (i.e. an acceptable probability of Type I error) and then use probability theory to determine the precise threshold which corresponds to that specified error rate. A general discussion of hypothesis tests at an introductory level can be found in textbooks of statistics such as DeGroot and Schervish (2002), or Moore and McCabe (2002); the standard advanced reference on the topic is Lehmann (1997).

Now, let us assume for the moment that we have only one object to consider. The hypothesis that the object is irrelevant is called the null hypothesis (and denoted by H_0), and the hypothesis that it is relevant is called the alternative hypothesis (H_1). The aim of the hypothesis test is to make a decision regarding the relevance of the object, that is, a decision as to which hypothesis should be accepted. Suppose the relevance score for the object under consideration is t. A decision regarding the relevance of the object is then made as follows:

1. Specify an acceptable level of Type I error p^*.
2. Use the sampling distribution of the relevance score under the null hypothesis to compute a threshold score corresponding to p^*. Let this threshold score be denoted by c.
3. If $t \geq c$, reject the null hypothesis and regard the object as relevant. If $t < c$, regard the object as irrelevant.

The specified error level p^* is called the significance level of the test and the corresponding threshold c the critical value.

Hypothesis testing can alternatively be thought of as a procedure by which relevance scores are converted into corresponding error probabilities. The null sampling distribution can be used to compute the probability p of making a Type I error if the threshold is set at exactly t, i.e. just low enough to select the given object. This then allows us to assert that the probability of obtaining a false positive if the given object is to be selected is at least p. This latter probability of Type I error is called a P-value. In contrast to relevance scores, P-values, being probabilities, have a clear interpretation. For instance, if we found that an object had a t-statistic value of 3 (say), it would be hard to tell whether the object should be regarded as relevant or not. However, if we found the corresponding P-value was 0.001, we would know that if the threshold were set just low enough to include the object, the false positive rate would be 1 in 1000, a fact that is far easier to interpret.

MAIN THRUST

We have seen that in the case of a single variable, relevance scores obtained from test statistics can be easily converted into error probabilities called P-values. However, practical data mining tasks, such as mining microarrays or consumer records, tend to be on a very large scale, with thousands, even millions of objects under consideration. Under these conditions of multiplicity, the conventional P-value described above no longer corresponds to the probability of obtaining a false positive.

An example will clarify this point. Consider once again the microarray analysis scenario, and assume that a suitable relevance scoring function has been chosen. Now, suppose we wish to set a threshold corresponding to a false positive rate of 0.05. Let the relevance score whose P-value is 0.05 be denoted by t_{05}. Then, in the case of a single variable/gene, if we were to set the threshold at t_{05}, the probability of obtaining a false positive would be 0.05. However, in the multi-gene setting, it is *each* of the thousands of genes under study that is effectively subjected to a hypothesis test with the specified error probability of 0.05. Thus, the chance of obtaining a false positive is no longer 0.05, but much higher. For instance, if each of 10000 genes were statistically independent, $(0.05 \times 10000) = 500$ genes would be mistakenly selected on average! In

effect, the very threshold which implied a false positive rate of 0.05 for a single gene now leaves us with hundreds of false positives.

Multiple hypothesis testing procedures address the issue of multiplicity in hypothesis tests and provide a way of setting appropriate thresholds in multi-variable problems. The remainder of this Section describes two well-known multiple testing methods (the Bonferroni and False Discovery Rate methods), and discusses their advantages and disadvantages.

Table 1 summarizes the numbers of objects in various categories, and will prove useful in clarifying some of the concepts presented below. The total number of objects under consideration is m, of which m_1 are relevant and m_0 are irrelevant. A total of S objects are selected, of which S_1 are true positives and S_0 are false positives. We follow the convention that variables relating to irrelevant objects have the subscript "0" (to signify the null hypothesis) and those relating to relevant objects the subscript "1" (for the alternative hypothesis). Note also that fixed quantities (e.g. the total number of objects) are denoted by lower-case letters, while variable quantities (e.g. the number of objects selected) are denoted by upper-case letters.

The initial stage of a multi-variable analysis follows from our discussion of basic hypothesis testing and is common to the methods discussed below. The data is processed as follows:

1. Score each of the m objects under consideration using a suitably chosen test statistic f. If the data corresponding to object j is denoted by D_j, and the corresponding relevance score by T_j:

   $$T_j = f(D_j)$$

Let the scores so obtained be denoted by T_1, T_2,…T_m.

Table 1. Summary table for multiple testing, following Benjamini and Hochberg (1995)

	Selected	Not selected	Total
Irrelevant objects	S_0	m_0-S_0	m_0
Relevant objects	S_1	m_1-S_1	m_1
Total	S	m-S	m

2. Convert each score T_j to a corresponding P-value P_j by making use of the relevant null sampling distribution

   $$T_j \xrightarrow{\text{null sampling distribution}} P_j$$

These P-values are called 'nominal P-values' and represent *per-test* error probabilities. The procedure by which a P-value corresponding to an observed test statistic is computed is not described here, but can be found in any textbook of statistics, such as those mentioned previously. Essentially, the null cumulative distribution function of the test statistic is used to compute the probability of Type I error at threshold T_j.

3. Arrange the P-values obtained in the previous step in ascending order (smaller P-values correspond to objects more likely to be relevant). Let the *ordered* P-values be denoted by:

   $$P_{(1)} \le P_{(2)} \le \cdots \le P_{(m)}$$

Thus, each object has a corresponding P-value. In order to select a subset of objects, it will therefore be sufficient to determine a threshold in terms of nominal P-value.

The Bonferroni Method

This is perhaps the simplest method for correcting P-values. We first specify an acceptable probability of committing at least one Type I error, and then calculate a corresponding threshold in terms of nominal P-value.

Procedure

1. Specify an acceptable probability p^* of at least one Type I error being committed.
2. Recall that $P_{(1)} \le P_{(2)} \le \ldots \le P_{(m)}$ represent the ordered P-values. Find the largest i which satisfies the following inequality: $P_{(i)} \le \frac{p^*}{m}$

   Let the largest i found be denoted k.

3. Select the k objects corresponding to P-values $P_{(1)}$, $P_{(2)}$…$P_{(k)}$ as relevant.

If we assume that all *m* tests are statistically independent, it is easy to show that this simple procedure does indeed guarantee that the probability of obtaining at least one false positive is *p**.

Advantages

- The basic Bonferroni procedure is extremely simple to understand and use.

Disadvantages

- The Bonferroni procedure sets the threshold to meet a specified probability of making *at least one* Type I error. This notion of error is called "Family Wise Error Rate", or FWER, in the statistical literature. FWER is a very strict control of error, and is far too conservative for most practical applications. Using FWER on datasets with large numbers of variables often results in the selection of a *very* small number of objects, or even none at all.
- The assumption of statistical independence between tests is a strong one, and almost never holds in practice.

The FDR Method of Benjamini and Hochberg

An alternative way of thinking about errors in multiple testing is the false discovery rate **(FDR)** (Benjamini & Hochberg, 1995). Looking at Table 1 we can see that of *S* objects selected, S_0 are false positives. FDR is simply the average proportion of false positives among the objects seleced:

$$FDR \equiv E\left[\frac{S_0}{S}\right]$$

Where, E[] denotes expectation. The Benjamini and Hochberg method allows us to compute a threshold corresponding to a specified FDR.

Procedure

1. Specify an acceptable FDR q*
2. Find the largest *i* which satisfies the following inequaity:

$$P_{(i)} \leq \frac{i}{m} \times q^*$$

Let the largest *i* found be denoted *k*.

(3) Select the *k* objects corresponding to P-values $P_{(1)}, P_{(2)} \ldots P_{(k)}$ as relevant.

Under the assumption that the *m* tests are statistically independent, it can be shown that this procedure leads to a selection of objects such that the FDR is indeed *q**.

Advantages

- The FDR concept is far less conservative than the Bonferroni method described above.
- Specifying an acceptable proportion *q** of false positives is an intuitively appealing way of controlling the error rate in object selection.
- The FDR method tells us in advance that a certain proportion of results are likely to be false positives. As a consequence, it becomes possible to plan for the associated costs.

Disadvantages

- Again, the assumption of statistically independent tests rarely holds.

FUTURE TRENDS

A great deal of recent work in statistics, machine learning and data mining has focused on various aspects of multiple testing in the context of object selection. Some important areas of active research which were not discussed in detail are briefly described below.

Robust FDR Methods

The FDR procedure outlined above, while simple and computationally efficient, makes several strong assumptions, and while better than Bonferroni is still often too conservative for practical problems (Storey & Tibshirani, 2003). The recently introduced "q-value" (Storey, 2003) is a more sophisticated approach to FDR correction and provides a very robust methodol-

ogy for multiple testing. The q-value method makes use of the fact that P-values are uniformly distributed under the null hypothesis to accurately estimate the FDR associated with a particular threshold. Estimated FDR is then used to set an appropriate threshold. The q-value approach is an excellent choice in many multivariable settings.

Resampling Based Methods

In recent years a number of resampling based methods have been proposed for multiple testing (see e.g. Westfall & Young, 1993). These methods make use of computationally intensive procedures such as the bootstrap (Efron, 1982; Davison & Hinckley, 1997) to perform non-parametric P-value corrections. Resampling methods are extremely powerful and make fewer strong assumptions than methods based on classical statistics, but in many data mining applications the computational burden of resampling may be prohibitive.

Machine Learning of Relevance Functions

The methods described in this chapter allowed us to determine threshold relevance scores, but very little attention was paid to the important issue of choosing an appropriate relevance scoring function. Recent research in bioinformatics (Broberg, 2003; Mukherjee, 2004a) has shown that the effectiveness of a scoring function can be very sensitive to the statistical model underlying the data, in ways which can be difficult to address by conventional means. When fully labeled data (i.e. datasets with objects flagged as relevant/irrelevant) are available, canonical supervised algorithms (see e.g. Hastie, Tibshirani, & Friedman , 2001) can be used to learn effective relevance functions. However, in many cases - microarray data being one example - fully labeled data is hard to obtain. Recent work in machine learning (Mukherjee, 2004b) has addressed the problem of learning relevance functions in an unsupervised setting by exploiting a probabilistic notion of stability; this approach turns out to be remarkably effective in settings where underlying statistical models are poorly understood and labeled data unavailable.

CONCLUSION

An increasing number of problems in industrial and scientific data analysis involve multiple testing, and there has consequently been an explosion of interest in the topic in recent years. This Chapter has discussed a selection of important concepts and methods in multiple testing; for further reading we recommend Benjamini and Hochberg (1995) and Dudoit, Shaffer, & Boldrick, (2003).

REFERENCES

Benjamini, Y., & Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *J. Roy. Stat. Soc. B 57*, 289-300.

Broberg, P. (2003). Statistical methods for ranking differentially expressed genes. *Genome Biology, 4*(6).

Cui, X., & Churchill, G. (2003). Statistical tests for differential expression in cDNA microarray experiments. *Genome Biology, 4*(210).

Davison, A. C., & Hinckley, D. V. (1997). *Bootstrap methods and their applications*. Cambridge University Press.

DeGroot, M. H., & Schervish, M. J. (2002). *Probability and statistics* (3rd ed.). Addison-Wesley.

Dudoit, S., Shaffer, J. P., & Boldrick, J. C. (2003). Multiple hypothesis testing in microarray experiments. *Statistical Science, 18*(1), 71-103.

Efron, B. (1982). *The jacknife, the bootstrap, and other resampling plans.* Society for Industrial and Applied Mathematics.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning*. Springer-Verlag.

Lehmann, E. L. (1997). *Testing statistical hypotheses* (2nd ed.). Springer-Verlag.

Moore, D. S., & McCabe, G. P. (2002). *Introduction to the practice of statistics* (4th ed.). W. H. Freeman.

Mukherjee, S. (2004a). A theoretical analysis of gene selection. *Proceedings of the IEEE Computer Society Bioinformatics Conference 2004 (CSB 2004)*. IEEE press.

Mukherjee, S. (2004b). *Unsupervised learning of ranking functions for high-dimensional data* (Tech. Rep. No. PARG-04-02). University of Oxford, Department of Engineering Science.

Piatetsky-Shapiro, G., & Tamayo, P. (2003). Microarray Data mining: Facing the Challenges. *SIGKDD Explorations, 5*(2).

Storey, J. D. (2003). The positive false discovery rate: A Bayesian interpretation and the q-value. *Annals of Statistics, 31*, 2013-2035.

Storey, J. D., & Tibshirani, R. (2003). Statistical significance for genome-wide studies. *Proceedings of the National Academy of Sciences, 100* (pp. 9440-9445).

Westfall, P. H., & Young, S. S. (1993). *Resampling-based multiple testing: Examples and methods for p-value adjustment*. John Wiley & Sons.

KEY TERMS

Alternative Hypothesis: The hypothesis that an object is relevant. In general terms, the alternative hypothesis refers to the set of events that is the complement of the null hypothesis.

False Discovery Rate (FDR): The expected proportion of false positives among the objects selected as relevant.

False Negative: The error committed when a truly relevant object is not selected. More generally, a false negative occurs when the null hypothesis is erroneously accepted. Also called Type II error.

False Positive: The error committed when an object is selected as relevant when it is in fact irrelevant. More generally, a false positive occurs when the null hypothesis is erroneously rejected. Also called Type I error.

Gene Microarray Data: Measurements of mRNA abundances derived from biochemical devices called microarrays. These are essentially measures of gene activity.

Hypothesis Test: A formal statistical procedure by which an interesting hypothesis (the alternative hypothesis) is accepted or rejected on the basis of data.

Multiple Hypothesis Test: A formal statistical procedure used to account for the effects of multiplicity in a hypothesis test.

Null Hypothesis: The hypothesis that an object is irrelevant. In general terms, it is the hypothesis we wish to falsify on the basis of data.

P-Value: The P-value for an object is the probability of obtaining a false positive if the threshold is set just high enough to include the object among the set selected as relevant. More generally, it is the false positive rate corresponding to an observed test statistic.

Random Variable: A variable characterized by random behavior in assuming its different possible values.

Sampling Distribution: The distribution of values obtained by applying a function to random data.

Test Statistic: A relevance scoring function used in a hypothesis test. Classical test statistics (such as the t-statistic) have null sampling distributions which are known *a priori*. However, under certain circumstances null sampling distributions for arbitrary functions can be obtained by computational means.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 523-528, copyright 2005 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

Music Information Retrieval

Alicja A. Wieczorkowska
Polish-Japanese Institute of Information Technology, Poland

INTRODUCTION

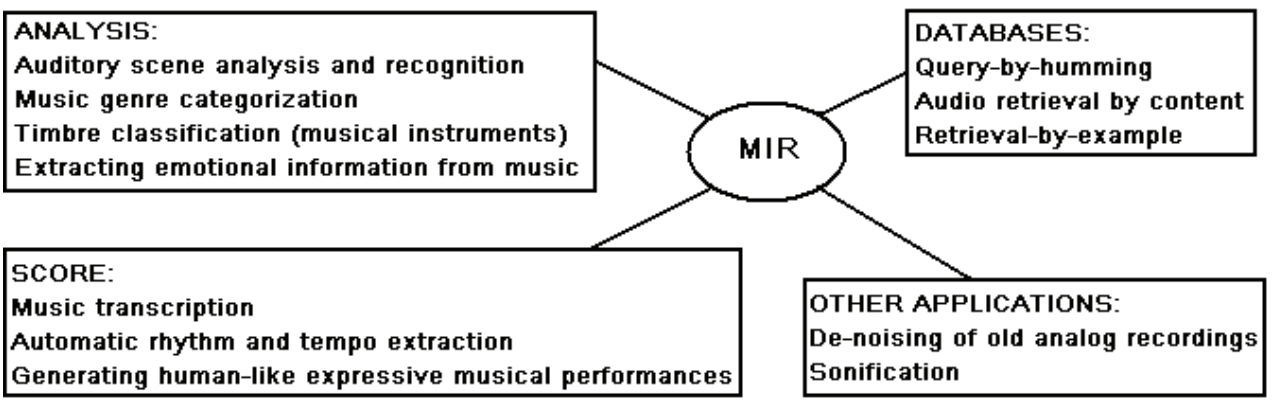
Music information retrieval (MIR) is a multi-disciplinary research on retrieving information from music, see Fig. 1. This research involves scientists from traditional, music and digital libraries, information science, computer science, law, business, engineering, musicology, cognitive psychology and education (Downie, 2001).

BACKGROUND

Huge amount of audio resources, including music data, is becoming available in various forms, both analog and digital. Notes, CDs, and digital resources of the World Wide Web are constantly growing in amount, but the value of music information depends on how easy it can be found, retrieved, accessed, filtered and managed. MIR consists in quick and efficient searching for various types of audio data of interest to the user, and filtering them in order to receive only the data items which satisfy the user's preferences (Fingerhut, 1997), (International Organization for Standardization, 2003), (Wieczorkowska & Ras, 2003). MIR applications include retrieval of music pieces from huge audio databases, and automatic production of music score on the basis of the presented input. The MIR topics are interrelated, since similar techniques can be applied for various purposes, e.g., source separation is applied in auditory scene analysis, music transcription, and even for restoring (de-noising) of all recordings. Generally, the research within MIR domain is focused on: harmonic structure analysis, note extraction, melody and rhythm tracking, timbre and instrument recognition, classification of type of the signal (speech, music, pitched vs. non-pitched), etc.

The research basically uses digital audio recordings, where sound waveform is digitally stored as a sequence of discrete samples representing the sound intensity at given time instant, and MIDI files, storing information on parameters of electronically synthesized sounds (voice, note on, note off, pitch bend etc.). Sound analysis and data mining tools are used to extract information from music files, in order to provide the data that meet user's needs (Wieczorkowska & Ras, 2001).

Figure 1. Topics of MIR



MAIN THRUST OF THE CHAPTER

Various types of music data are investigated in MIR research. Basic techniques of digital sound analysis come from speech processing, focused on automatic speech recognition and speaker identification (Foote, 1999). The obtained sound descriptors can be applied to facilitate content-based searching of music databases.

The issue of representation of music and multimedia information in a form that allows interpretation of the information's meaning is addressed by MPEG-7 standard, named Multimedia Content Description Interface. MPEG-7 provides a rich set of standardized tools to describe multimedia content through metadata, i.e. data about data, and music information description has been also taken into account in this standard (International Organization for Standardization, 2003).

The following topics are investigated within MIR domain:

- Auditory scene analysis and recognition (Rosenthal & Okuno, 1998), which focuses on various aspects of music, like timbre description, sound harmonicity, spatial origin, source separation, etc. (Bregman, 1990), based on signal processing techniques. Timbre is defined subjectively as this feature of sound that distinguishes two sounds of the same pitch, loudness and duration. Therefore, subjective listening tests are often performed in this research. One of the main topics of computational auditory scene analysis is automatic separation of individual sound sources from a mixture. It is difficult with mixtures of harmonic instrument sounds, where spectra overlap. However, assuming time-frequency smoothness of the signal, sound separation can be performed, and when sound changes in time are observed, onset, offset, amplitude and frequency modulation have similar shapes for all frequencies in the spectrum, thus a de-mixing matrix can be estimated for them (Virtanen, 2003), (Viste & Evangelista, 2003). Audio source separation techniques can also be used to source localization for auditory scene analysis. These techniques, like independent component analysis (ICA), originate from speech recognition in cocktail party environment, where many sound sources are present. ICA is used for finding underlying components from multidimensional statistical data, and it looks for components that are statistically independent (Vincent, Rodet, Röbel, Févotte, Carpentier, Gribonval, Benaroya, & Bimbot, 2003). Also, computational auditory scene recognition may aim at classifying auditory scenes into predefined classes, using audio information only. Examples of auditory scenes are various outside and inside environments, like streets, restaurants, offices, homes, cars etc. Statistical and nearest neighbor algorithms can be applied for this purpose. In the nearest neighbor algorithm the class (type of auditory scene in this case) is assigned on the basis of the distance of the investigated sample to the nearest sample, for which the class membership is known. Various acoustic features, based on Fourier spectral analysis (i.e. mathematic transform, decomposing the signal into frequency components), can be applied to parameterize the auditory scene for classification purposes. Effectiveness of this research approaches 70% correctness for about 20 auditory scenes (Peltonen, Tuomi, Klapuri, Huopaniemi, & Sorsa, 2002).
- Music genre categorization is aimed at automatic classification of music into various genres. This can be especially useful for large audio collections, if they are not manually labelled (Guaus & Herrera, 2006).
- Audio retrieval-by-example for orchestral music aims at searching for acoustic similarity in an audio collection, based on analysis of the audio signal. Given an example audio document, other documents in a collection can be ranked by similarity on the basis of long-term structure, specifically the variation of soft and louder passages, determined from envelope of audio energy versus time in one or more frequency bands (Foote, 2000). This research is a branch of audio retrieval by content. Audio query-by-example search can be also performed within a single document, when searching for sounds similar to the selected sound event. Such a system for content-based audio retrieval can be based on a self-organizing feature map, i.e. a special kind of a neural network, designed by analogy with a simplified model of the neural connections in the brain, and trained to find relationships in the data. Perceptual similarity can be assessed on the basis of spectral evolution, in order to find sounds of similar timbre (Spevak

& Polfreman, 2001). Neural networks are also used in other forms in audio information retrieval systems. For instance, time-delayed neural networks, i.e. neural nets with time delay inputs, are applied, since they perform well in speech recognition applications (Meier, Stiefelhagen, Yang, & Waibel, 2000). One of applications of audio retrieval-by-example is searching for the piece in a huge database of music pieces, with the use of so called audio fingerprinting – technology that allows piece identification. Given a short passage transmitted for instance via car phone, the piece is extracted, and, what is most important, also is extracted the information on the performer and title, linked to this piece in the database. In this way, the user may identify the piece of music with very high accuracy (95%), only on the basis of a small recorded (possibly noisy) passage.

- Transcription of music, defined as to writing down the musical notation for the sounds that constitute the investigated piece of music. Onset detection based on incoming energy in frequency bands, and multi-pitch estimation based on spectral analysis may be used as the main elements of an automatic music transcription system. The errors in such system may contain additional inserted notes, omissions, or erroneous transcriptions (Klapuri, Virtanen, Eronen, & Seppänen, 2001). Pitch tracking, i.e. estimation of pitch of note events in a melody or a piece of music, is often performed in many MIR systems. For polyphonic music, polyphonic pitch-tracking and timbre separation in digital audio is performed, with such applications as score-following and de-noising of old analog recordings. Wavelet analysis can be applied for this purpose, since it decomposes the signal into time-frequency space, and then musical notes can be extracted from the result of this decomposition (Popovic, Coifman, & Berger, 1995). Simultaneous polyphonic pitch and tempo tracking, aiming at automatic inferring a musical notation that lists the pitch and the time limits of each note, is a basis of the automatic music transcription. A musical performance can be modeled for these purposes using dynamic Bayesian networks, i.e. directed graphical models of stochastic processes (Cemgil, Kappen, & Barber, 2003). It is assumed that the observations may be generated by a hidden process that cannot be directly experimentally observed, and dynamic Bayesian networks represent the hidden and observed states in terms of state variables, which can have complex interdependencies. Dynamic Bayesian networks generalize hidden Markov models, which have one hidden node and one observed node per observation time.
- Automatic characterizing of rhythm and tempo of music and audio, revealing tempo and the relative strength of particular beats, is a branch of research on automatic music transcription. Since highly structured or repetitive music has strong beat spectrum peaks at the repetition times, it allows tempo estimation and distinguishing between different kinds of rhythms at the same tempo. The tempo can be estimated using beat spectral peak criterion (the lag of the highest peak exceeding assumed time threshold), accurately to within 1% in the analysis window (Foote & Uchihashi, 2001).
- Automatic classification of musical instrument sounds, aiming at accurate identification of musical instruments playing in a given recording, based on various sound analysis and data mining techniques (Herrera, Amatriain, Batlle, & Serra, 2000), (Wieczorkowska, 2001), (Eronen, 2001), (Aniola & Lukasik, 2007). This research is mainly focused on monophonic sounds, and sound mixes are usually addressed in the research on separation of sound sources (Kostek, Dziubiński, & Dalka, 2005). In most cases, sounds of instruments of definite pitch have been investigated, but recently also research on percussion is undertaken. Various analysis methods are used to parameterize sounds for instrument classification purposes, including time-domain description, Fourier and wavelet analysis. Classifiers range from statistic and probabilistic methods, through learning by example, to artificial intelligence methods. Effectiveness of this research ranges from about 70% accuracy for instrument identification, to more than 90% for instruments family (i.e. strings, winds etc.), approaching 100% for discriminating impulsive and sustained sounds, thus even exceeding human performance. Such instrument sound classification can be included in automatic music transcription systems.
- Query-by-humming systems, which search melodic databases using sung queries (Adams, Bartsch, & Wakefield, 2003). This topic rep-

resents audio retrieval by contents. Melody is usually coarsely quantized with respect to pitch and duration, assuming moderate singing abilities of users. Music retrieval system takes such an aural query (a motif, or a theme) as input, and searches the database for the piece this query comes from. Markov models, based on Markov chains, can be used for modeling musical performances. Markov chain is a stochastic process for which the parameter is discrete time values. In Markov sequence of events the probability of future states depends on the present state; in this case, states represent pitch (or set of pitches) and duration (Birmingham, Dannenberg, Wakefield, Bartsch, Bykowski, Mazzoni, Meek, Mellody, & Rand, 2001). Query-by-humming is one of more popular topics within MIR domain.

- Generating human-like expressive musical performances, with appropriately adjusted dynamics (i.e. loudness), rubato (variation in notes' length), vibrato (changes of pitch) etc. (Mantaras & Arcos, 2002).
- Extracting emotional information from music - estimating what mood is envoked by a given piece of music (tenderness, sadness, joy, calmness etc.). Also, if the musical content can be changed, the mood of the piece can be changed too, thus creating a different version of this piece. This research can be also applied for psychological, or even medical purposes (Wieczorkowska, Synak, & Raś, 2006), (Koelsch, Fritz, Cramon, Müller, & Friederici, 2006).
- Sonification, in which utilities for intuitive auditory display (i.e. in audible form) are provided through a graphical user interface (Ben-Tal, Berger, Cook, Daniels, Scavone, & Cook, 2002).

The topics mentioned above interrelate and sometimes partially overlap. For instance, auditory scene analysis and recognition take into account broad range of recordings, containing numerous acoustic elements to identify and analyze. Query by humming requires automatic transcription of music, since the input audio samples must be first transformed into the form based on musical notation, describing basic melodic features of the query. Audio retrieval by example and automatic classification of musical instrument sounds are both branches of retrieval by content. Transcription of music requires pitch tracking, and also automatic characterizing of rhythm and tempo. Pitch tracking is needed in many areas, including music transcription, query by humming, and even automatic classification of musical instrument sounds, since pitch is one of the features characterizing instrumental sound. Sonification and generating human-like expressive musical performances are both related to sound synthesis, needed to create auditory display or emotional performance. All these topics are focused on broad domain of music and its aspects.

Results of MIR research are not always easily measurable, since they are usually validated via subjective tests. Other topics, like transcription of music, may produce errors of various importance (wrong pitch, length, omission etc.), and comparison of the obtained transcript with the original score can be measured in many ways, depending on the considered criteria. The easiest estimation and comparison of results can be performed in case of recognition of singular sound events or files. In case of query by example, very high recognition rate has been already obtained (95% of correct piece identification via audio fingerprinting), reaching commercial level.

The research on MIR is gaining an increasing interest from the scientific community, and investigation of further issues in this domain can be expected.

FUTURE TRENDS

Multimedia databases and library collections need efficient tools for content based search. Therefore, we can expect intensification of research effort on MIR, which may aid searching music data. Especially, tools for query-by-example and query-by-humming are needed, tools for automatic music transcription, and also for extracting or changing emotions in music, so these areas should be broadly investigated in the nearest future.

CONCLUSION

MIR has a broad range of research and applications, including audio retrieval by content, automatic music transcription etc. Results of this research help users in finding the audio data they need, even if the users itself are not experienced musicians. Constantly growing, huge audio resources evoke demand for efficient

tools to deal with this data, therefore MIR becomes a dynamically developing field of research.

ACKNOWLEDGMENT

This work was supported by the Research Center of Polish-Japanese Institute of Information Technology, supported by the Polish National Committee for Scientific Research (KBN).

REFERENCES

Aniola, P., & Lukasik, E. (2007, May). *JAVA library for automatic musical instruments recognition.* AES 122 Convention, Vienna, Austria

Adams, N. H., Bartsch, M. A., & Wakefield, G. H. (2003, October). Coding of sung queries for music information retrieval. *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics WASPAA'03*. New Paltz, NY.

Ben-Tal, O., Berger, J., Cook, B., Daniels, M., Scavone, G., & Cook, P. (2002, July). SONART: The Sonification Application Research Toolbox. *Proceedings of the 2002 International Conference on Auditory Display*. Kyoto, Japan, 151-153.

Birmingham, W. P., Dannenberg, R. D., Wakefield, G. H., Bartsch, M. A., Bykowski, D., Mazzoni, D., Meek, C., Mellody, M., & Rand, B. (2001, October). MUSART: Music retrieval via aural queries. *Proceedings of ISMIR 2001, 2nd Annual International Symposium on Music Information Retrieval*. Bloomington, Indiana, 73-81.

Bregman, A. S. (1990). *Auditory scene analysis, the perceptual organization of sound*. MIT Press.

Cemgil, A. T., Kappen, B., & Barber, D. (2003, October). Generative Model Based Polyphonic Music Transcription. *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics WASPAA'03*. New Paltz, NY.

Downie, J. S. (2001, October). Wither music information retrieval: ten suggestions to strengthen the MIR research community. In J. S. Downie & D. Bainbridge (Eds.), *Proceedings of the Second Annual International Symposium on Music Information Retrieval: ISMIR 2001*. Bloomington, Indiana, 219-222.

Eronen A. (2001, October) Comparison of features for musical instrument recognition. *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics WASPAA 2001* New Paltz, NY

Fingerhut, M. (1997). Le multimédia dans la bibliothèque. *Culture et recherche 61*.

Foote, J. (1999). An Overview of Audio Information Retrieval. *Multimedia Systems,. 7*(1), 2-11. ACM Press/Springer.

Foote, J. (2000, October). ARTHUR: Retrieving Orchestral Music by Long-Term Structure. *Proceedings of the International Symposium on Music Information Retrieval ISMIR 2000*. Plymouth, Massachusetts.

Foote, J. & Uchihashi, S. (2001, August). The Beat Spectrum: A New Approach to Rhythm Analysis. *Proceedings of the International Conference on Multimedia and Expo ICME 2001*. Tokyo, Japan, 1088-1091.

Guaus, E. & Herrera, P. (2006, October). Music Genre Categorization in Humans and Machines, *AES 121st Convention*, San Francisco.

Herrera, P., Amatriain, X., Batlle, E., & Serra X. (2000, October). Towards instrument segmentation for music content description: a critical review of instrument classification techniques. *Proceedings of the International Symposium on Music Information Retrieval ISMIR 2000, Plymouth, MA.*

International Organization for Standardization ISO/IEC JTC1/SC29/WG11 (2003). *MPEG-7 Overview*. The Internet <http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm>

Koelsch, S., Fritz, T., Cramon, D. Y. v., Müller, K, & Friederici, A. D. (2006). Investigating Emotion With Music: An fMRI Study. *Human Brain Mapping, 27*, 239-250.

Klapuri, A., Virtanen, T., Eronen, A., & Seppänen, J. (2001, September). Automatic transcription of musical recordings. *Proceedings of the Consistent & Reliable Acoustic Cues for sound analysis CRAC Workshop*. Aalborg, Denmark.

Kostek, B., Dziubiński, M. & Dalka, P. (2005) Estimation of Musical Sound Separation Algorithm Eÿectiveness Employing Neural Networks. *Journal of Intelligent Information Systems 24*, 133-157

Mantaras, R. L. de, & Arcos, J. L. (Fall 2002). AI and Music. From Composition to Expressive Performance. *AI Magazine*, 43-58.

Meier, U., Stiefelhagen, R., Yang, J., & Waibel, A. (2000). Towards Unrestricted Lip Reading. *International Journal of Pattern Recognition and Artificial Intelligence,14*(5), 571-586.

Peltonen, V., Tuomi, J., Klapuri, A., Huopaniemi, J., & Sorsa, T. (2002, May). Computational Auditory Scene Recognition. *International Conference on Acoustics Speech and Signal Processing*. Orlando, Florida.

Popovic, I., Coifman, R., & Berger, J. (1995). Aspects of Pitch-Tracking and Timbre Separation: Feature Detection in Digital Audio Using Adapted Local Trigonometric Bases and Wavelet Packets. Center for Studies in Music Technology, Yale University, Research Abstract.

Rosenthal, D. & Okuno, H. G., (Eds.) (1998). *Computational Auditory Scene Analysis. Proceedings of the IJCAI-95 Workshop*. Lawrence Erlbaum Associates, Mahwah, New Jersey.

Spevak, C. & Polfreman, R. (2001, October). Sound Spotting – a Frame-Based Approach. In J. S. Downie & D. Bainbridge (Eds.), *Proceedings of the Second Annual International Symposium on Music Information Retrieval: ISMIR 2001*. Bloomington, Indiana, 35-36.

Vincent, E., Rodet, X., Röbel, A., Févotte, C., Carpentier, É. L., Gribonval, R., Benaroya, L., & Bimbot, F. (2003, April). A tentative typology of audio source separation tasks. *Proceedings of the 4th Symposium on Independent Component Analysis and Blind Source Separation*. Nara, Japan, 715-720.

Virtanen, T. (2003, September). Algorithm for the separation of harmonic sounds with time-frequency smoothness constraint. *Proceedings of the 6th International Conference on Digital Audio Effects DAFX-03*. London, UK.

Viste, H., & Evangelista, G. (2003, October). Separation of Harmonic Instruments with Overlapping Partials in Multi-Channel Mixtures. *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics WASPAA'03*, New Paltz, NY.

Wieczorkowska, A. (2001). Musical Sound Classification based on Wavelet Analysis, *Fundamenta Informaticae Journal 47*, 1/2, 175-188.

Wieczorkowska, A. & Ras, Z. (2001). Audio content description in sound databases. In Zhong, N., Yao, Y., Liu, J., & Ohsuga, S. (Eds.), *Web Intelligence: Research and Development, LNCS/LNAI 2198*, Springer, 175-183.

Wieczorkowska, A. & Ras, Z. W. (Eds.) (2003). Music Information Retrieval. Special Issue, Journal *of Intelligent Information Systems 21*, 1, Kluwer.

Wieczorkowska, A., Synak, P., & Raś, Z. W. (2006). Multi-Label Classification of Emotions in Music. In Kłopotek, M. A., Wierzchoń, S. T., Trojanowski, K. (Eds), *Intelligent Information Processing and Web Mining*. Springer, Advances in Soft Computing, 307-315.

KEY TERMS

Digital Audio: Digital representation of sound waveform, recorded as a sequence of discrete samples, representing the intensity of the sound pressure wave at given time instant. Sampling frequency describes the number of samples recorded in each second, and bit resolution describes the number of bits used to represent the quantized (i.e. integer) value of each sample.

Fourier Analysis: Mathematical procedure for spectral analysis, based on Fourier transform that decomposes a signal into sine waves, representing frequencies present in the spectrum.

Information Retrieval: The actions, methods and procedures for recovering stored data to provide information on a given subject.

Metadata: Data about data, i.e. information about the data.

MIDI: Musical Instrument Digital Interface. MIDI is a common set of hardware connectors and digital codes, used to interface electronic musical instruments and other electronic devices. MIDI controls actions such as note events, pitch bends, etc. while the sound is generated by the instrument itself.

Music Information Retrieval: Multi-disciplinary research on retrieving information from music.

Pitch Tracking: Estimation of pitch of note events in a melody or a piece of music.

Sound: A physical disturbance in the medium through which it is propagated. Fluctuation may change routinely and such a periodic sound is perceived as having pitch. The audible frequency range is from about 20 Hz (hertz, i.e. cycles per second) to about 20 kHz. Harmonic sound wave consists of frequencies being integer multiples of the first component (fundamental frequency), corresponding to the pitch. The distribution of frequency components is called spectrum. Spectrum and its changes in time can be analyzed using mathematical transforms, such as Fourier or wavelet transform.

Wavelet Analysis: Mathematical procedure for time-frequency analysis, based on wavelet transform that decomposes a signal into shifted and scaled versions of the original function called wavelet.

Neural Networks and Graph Transformations

Ingrid Fischer
University of Konstanz, Germany

INTRODUCTION

As the beginning of the area of artificial neural networks the introduction of the artificial neuron by McCulloch and Pitts is considered. They were inspired by the biological neuron. Since then many new networks or new algorithms for neural networks have been invented with the result. In most textbooks on (artificial) neural networks there is no general definition on what a neural net is but rather an example based introduction leading from the biological model to some artificial successors. Perhaps the most promising approach to define a neural network is to see it as a network of many simple processors ("units"), each possibly having a small amount of local memory. The units are connected by communication channels ("connections") that usually carry numeric (as opposed to symbolic) data called the weight of the connection. The units operate only on their local data and on the inputs they receive via the connections. It is typical of neural networks, that they have great potential for parallelism, since the computations of the components are largely independent of each other. Typical application areas are:

- Capturing associations or discovering regularities within a set of patterns;
- Any application where the number of variables or diversity of the data is very great;
- Any application where the relationships between variables are vaguely understood; or,
- Any application where the relationships are difficult to describe adequately with conventional approaches.

Neural networks are not programmed but can be trained in different ways. In supervised learning, examples are presented to an initialized net. From the input and the output of these examples, the neural net learns. There are as many learning algorithms as there are types of neural nets. Also learning is motivated physiologically. When an example is presented to a neural network it cannot recalculate, several different steps are possible: the neuron's data is changed, the connection's weight is changed or new connections and/or neurons are inserted. Introductory books into neural networks are (Graupe, 2007; Colen, Kuehn & Sollich, 2005).

There are many advantages and limitations to neural network analysis and to discuss this subject properly one must look at each individual type of network. Nevertheless there is one specific limitation of neural networks potential users should be aware of. Neural networks are more or less, depending on the different types, the ultimate "black boxes". The final result of the learning process is a trained network that provides no equations or coefficients defining a relationship beyond its own internal mathematics.

Graphs are widely used concepts within computer science, in nearly every field graphs serve as a tool for visualization, summarization of dependencies, explanation of connections, etc. Famous examples are all kinds of different nets and graphs as e.g. semantic nets, petri nets, flow charts, interaction diagrams or neural networks, the focus of this chapter. Invented 35 years ago, graph transformations have been constantly expanding. Wherever graphs are used, graph transformations are also applied (Rozenberg, 1997; Ehrig, Engels, Kreowski, and Rozenberg, 1999; Ehrig, Kreowski, Montanari, and Rozenberg, 1999; Ehrig, Ehrig, Prange & Taentzer, 2006).

Graph transformations are a very promising method for modeling and programming neural networks. The graph part is automatically given as the name "neural network" already indicates. Having graph transformations as methodology, it is easy to model algorithms on this graph structure. Structure preserving and structure changing algorithms can be modeled equally well. This is not the case for the widely used matrices programmed mostly in *C* or *C++*. In these approaches modeling structure change becomes more difficult.

This directly leads to a second advantage. Graph transformations have proven useful for visualizing

the network and its algorithms. Most modern neural network simulators have some kind of visualization tool. Graph transformations offer a basis for this visualization as the algorithms are already implemented in visual rules.

When having a formal methodology at hand, it is also possible to use it for proving properties of nets and algorithms. Especially in this area, earlier results for graph transformation systems can be used. Three possibilities are especially promising: first it is interesting whether an algorithm is terminating. Though this question is not decidable in the general case, the formal methods of graph rewriting and general rewriting offer some chances to prove termination for neural network algorithms. The same holds for the question whether the result produced by an algorithm is useful, whether the learning of a neural network was successful. Then it helps proving whether two algorithms are equivalent. Finally possible parallelism in algorithms can be detected and described based on results for graph transformation systems.

BACKGROUND

A Short Introduction to Graph Transformations

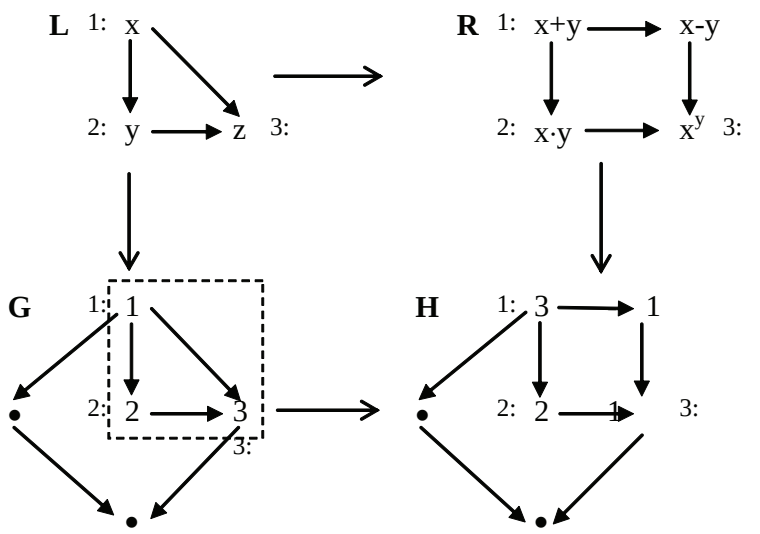
Despite the different approaches to handle graph transformations, there are some properties all approaches have in common. When transforming a graph *G* somehow, it is necessary to specify what part of the graph, what subgraph *L*, has to be exchanged. For this subgraph, a new graph *R* must be inserted. When applying such a rule to a graph *G* three steps are necessary:

- Choose an occurrence of *L* in *G*.
- Delete *L* from *G*.
- Insert *R* into the remainder of *G*.

In Figure 1 a sample application of a graph transformation rule is shown. The left hand side *L* consists of three nodes *(1:, 2:, 3:)* and three edges. This graph is embedded into a graph *G*. Numbers is *G* indicate how the nodes of *L* are matched. The embedding of edges is straightforward. In the next step *L* is deleted from *G* and *R* is inserted. If *L* is simply deleted from *G*, hanging edges remain. All edges ending/starting at *1:,2:,3:* are missing one node after deletion. With the help of numbers *1:,2:,3:* in the right hand side *R*, it is indicated how these hanging edges are attached to *R* inserted in *G/L*. The resulting graph is *H*.

Simple graphs are not enough for modeling real world applications. Among the different extensions two are of special interest. First graphs and graph rules can be labeled. When *G* is labeled with numbers, *L* is labeled with variables and *R* is labeled with terms over *L*'s variables. This way, calculations can be modeled. Taking our example and extending *G* with numbers *1,2,3*, the left hand side *L* with variables *x,y,z* and the

Figure 1. The application of a graph rewrite rule L → R to a graph G



right hand side with terms $x+y$, $x-y$, $x \cdot y$, x^y is shown in Figure 1. When L is embedded in G, the variables are set to the numbers of the corresponding nodes. The nodes in H are labeled with the result of the terms in R when the variable settings resulting from the embedding of L in G are used.

Also application conditions can be added restricting the application of a rule. For example the existence of a certain subgraph A in G can be allowed or forbidden. A rule can only be applied if A can be found resp. not found in G. Additionally label based application conditions are possible. Above rule could be extended by asking for $x < y$. Only in this case the rule would be applied.

Combining Neural Networks and Graph Transformations

Various proposals exist, how graph transformations and neural networks can be combined having different goals in mind. Several ideas originate from evolutionary computing (De Jong & Pollack, 2001; Curran & O'Riodan, 2002; Siddiqi & Lucas, 1998), others stem from electrical engineering as (Wan & Beaufays, 1998). The approach of (Fischer, 2000) has its roots in graph transformations itself. Despite these sources only three really different ideas can be found:

- In most of the papers (De Jong, and Pollack, 2001; Curran, and O'Riodan, 2002; Siddiqi, and Lucas 1998) some basic graph operators like *insert-node*, *delete-node*, *change-attribute*, etc. are used. This set of operators differs in the approaches. Additionally to these operators some kind of application condition exists stating which rule has to be applied when on which nodes or edges. These application conditions can be directed acyclic graphs giving the sequence of the rule applications. It can also be tree-based, where newly created nodes are handed to different paths in the tree. The main application area of these approaches is to grow neural networks from just one node. In this field also other grammar-based approaches can be found as (Cantu-Paz, and Kamath, 2002) where matrices are rewritten or (Browse, Hussain, and Smili,e 1999) taking attributed grammars. In (Tsakonas, and Dounias, 2002) feedforward neural networks are grown with the help of a grammar in Backus-Naur-Form.
- In (Wan, and Beaufays, 1998) signal flow graphs known from electrical engineering are used to model the information flow through the net. With the help of rewrite rules, the elements can be reversed, so that the signal flow is going in the opposite direction as before. This way gradient descent based training methods as backpropagation, a famous training algorithm (Rojas, 2000), can be derived. A disadvantage of this method is that no topology changing algorithms can be modeled.
- In (Fischer, 2000; Sona 2002) arbitrary (neural) nets are modeled as graphs and transformed by arbitrary transformation rules. Because this is the most general approach it will be explained in detail in the following sections.

MAIN THRUST OF THE CHAPTER

In the remainder of this paper one special sort of neural network, the so-called probabilistic neural networks together with training algorithms are explained in detail.

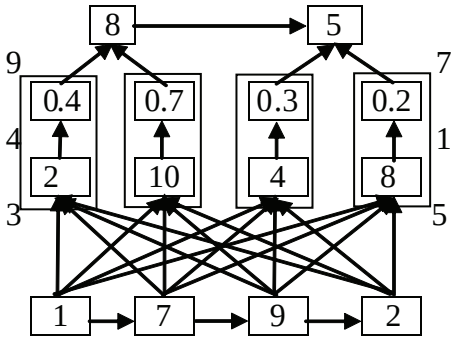
Probabilistic Neural Networks

The main purpose of a probabilistic neural network is to sort patterns into classes. It always has three layers, an input layer, a hidden layer and an output layer. The input neurons are connected to each neuron in the hidden layer. The neurons of the hidden layer are connected to one output neuron each. Hidden neurons are modeled with two nodes, as they have an input value and do some calculations on it resulting in an output value. Neurons and connections are labeled with values resp. weights. In Figure 2 a probabilistic neural network is shown.

Calculations in Probabilistic Neural Networks

First input is presented to the input neurons. The next step is to calculate the input value of the hidden neurons. The main purpose of the hidden neurons is to represent examples of classes. Each neuron represents one example. This example forms the weights from the input neurons to this special hidden neuron. When an input is presented to the net, each neuron in the hidden

Figure 2. A probabilistic neural network seen as graph



Note: Please note that not all edges are labeled due to space reasons

layer computes the probability, that it is the example it models. Therefore first the Euclidean distance between the activation of the input neurons and the weight of the connection to hidden layer's neuron is computed. The result coming via the connections is summed up to within a hidden neuron. This is the distance, the current input has from the example modeled by the neuron. If the exact example is inserted into the net, the result is *0*. In Figure 3 this calculation is shown in detail. The given graph transformation rule can be applied to the net shown in Figure 2. The labels *i* are variables modeling the input values of the input neurons, *w* models the weights of the connections. In the right hand side *R* a formula is given to calculate the input activation.

Then a Gaussian function is used to calculate the probability, that the inserted pattern is the example pattern of the neuron. The radius σ of the Gaussian function has to be chosen during the training of the net or it has to be adapted by the user. The result is the output activation of the neuron.

The main purpose of the connection between hidden and output layer is to sum up the activations of the hidden layer's neurons for one class. These values are multiplied with a weight associated to the corresponding connection. The output neuron with the heighest activation represents the class of the input pattern.

Training Neural Networks

For the probabilistic neural networks, different methods are used. The classical training method is the following: The complete net is built up by inserting one hidden neuron for each training pattern. As weights of the connections from the input neuron to the hidden neuron representing this example, the training pattern itself is taken. The connections from the hidden layer to the output layer are weighted with *1/m* if there are *m* training patterns for one class. σ, the radius of the Gaussian function used, is usually simply set to the average distance between centers of Gaussians modeling training patterns.

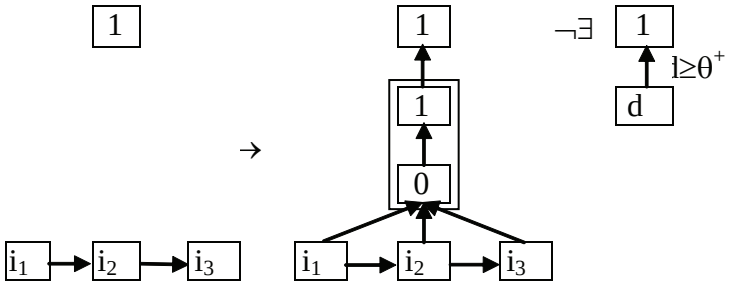
A more sophisticated training method called *Dynamic Decay Adjustment*. The algorithm also starts with no neuron in the hidden layer. If a pattern is presented to the net, first the activation of all existing hidden neurons is calculated. If there is a neuron whose activation is equal or higher to a given threshold θ^+, this neuron covers the input pattern. If this is not the case a new hidden neuron is inserted into the net as shown in Figure 4. With the help of this algorithm fewer hidden neurons are inserted.

FUTURE TRENDS

The area of graph grammars and graph transformation systems several programming environments, especially *AGG* and *Progress* (Ehrig, Engels, Kreowski, and Rozenberg, 1999) have been developed. They can be used to obtain new implementations of neural network algorithms, especially of algorithms, that change the net's topology.

Figure 3. Calculating the input activation of a neuron

Figure 4. A new neuron is inserted into a net (this example is taken from Dynamic Decay Adjustment)



Pruning means the deletion of neurons and connections between neurons in already trained neural networks to minimize their size. Several mechanisms exist, that mainly differ in the evaluation function determining the neurons/connections to be deleted. It is an interesting idea to model pruning with graph transformation rules.

A factor graph is a bipartite graph that expresses how a global function of many variables factors into a product of local functions. It subsumes other graphical means of artificial intelligence as Bayesian networks or Markov random fields. It would be worth checking whether graph transformation systems are helpful in formulating algorithms for these soft computing approaches. Also for more biological applications the use of graph transformation is interesting.

CONCLUSION

Modeling neural networks as graphs and algorithms on neural networks as graph transformations is an easy to use and straightforward method. It has several advantages. First, the structure of neural nets is supported. When modeling the net topology and topology changing algorithms, graph transformation systems can present their full power. This might be of special interest for educational purposes where it's useful to visualize step-by-step what algorithms do. Finally the theoretical background of graph transformation and rewriting systems offers several possibilities for proving termination, equivalence etc. of algorithms.

REFERENCES

Browse, R.A., Hussain, T.S., & Smillie, M.B. (1999). Using Attribute Grammars for the Genetic Selection of Backpropagation Networks for Character Recognition. In Nasrabadi, N., & Katsaggelos, A. (eds.) *Proceedings of Applications of Artificial Neural Networks in Image Processing IV*. San Jose, CA, USA, 26 – 34.

Cantu-Paz, E. & C. Kamath (2002). Evolving Neural Networks for the Classification of Galaxies, *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO 2002, Morgan Kaufmann Publishers, San Francisco, 1019-1026.

Curran, D., & O'Riordan, C. (2002). *Applying Evolutionary Computation to Designing Neural Networks: A Study of the State of the Art*. Technical Report of the Department of Information Technology, National University of Ireland, Galway, Ireland.

Colen, A. C. C., Kuehn, R., & Sollich, P. (2005) *Theory of Neural Information Processing Systems.* Oxford University Press.

De Jong, E, & Pollack, J. (2001). Utilizing Bias to Evolve Recurrent Neural Networks. *Proceedings of the International Joint Conference on Neural Networks*, volume 4. Renaissance Hotel, Washington, DC , USA, 2667 – 2672.

Ehrig, H., Engels, G., Kreowski, H.-J., & Rozenberg, G. (eds.) (1999). *Handbook on Graph Grammars and Computing by Graph Transformation*, Volume 2 *Applications, Languages and Tools*, World Scientific, Singapore.

Ehrig, H., Kreowski, H.-J., Montanari, U., & Rozenberg, G. (Editors) (1999). *Handbook on Graph Gram-*

mars and Computing by Graph Transformation, Volume 3 *Concurrency*, World Scientific, Singapore.

Ehrig, H., Ehrig, K., Prange, U., & Taentzer, G. (2006*) Fundamentals of Algebraic Graph Transformation* (Texts in Theoretical Computer Science EATCS). Springer.

Fischer, I. (2000). *Describing Neural Networks with Graph Transformations*, PhD Thesis, Computer Science, Friedrich-Alexander University Erlangen-Nuremberg.

Graupe, D. (2007*) Principles of Artificial Neural Networks*: World Scientific Publishing

Siddiqi, A. & Lucas, S.M. (1998). A comparison of matrix rewriting versus direct encoding for evolving neural networks. *IEEE International Conference on Evolutionary Computation*, Anchorage, Alaska, USA, 392 – 397.

Sona, D. (2002). *A Proposal for an Abstract Neural Machine*, Ph.D. Thesis, TD-8/02, Università di Pisa, Dip. Informatica,.

Nipkow, T., & Baader, F. (1999). *Term Rewriting and All That.* Cambridge University Press.

Rozenberg, G. (ed.) (1997). *Handbook of Graph Grammars and Computing by Graph Transformations*, Volume 1 *Foundations*, World Scientific, Singapore.

Tsakonas, A., & Dounias, D. (2002). A Scheme for the Evolution of Feedforward Neural Networks using BNF-Grammar Driven Genetic Programming. *EUNITE - EUropean Network on Intelligent TEchnologies for Smart Adaptive Systems*, Algarve, Portugal.

Klop, J.W., De Vrijer, R.C., & Bezem, M. (2003). *Term Rewriting Systems.* Cambridge University Press.

Wan, E. & Beaufays, F. (1998). Diagrammatic Methods for Deriving and Relating Temporal Neural Network Algorithms. In Giles, C. & Gori, M. (eds.) *Adaptive Processing of Sequences and Data Structures*, International Summer School on Neural Networks, Vietri sul Mare, Salerno, Italy, Lecture Notes in Computer Science 1387, 63 – 98.

KEY TERMS

Confluence: A rewrite system is confluent, if no matter in which order rules are applied, they lead to the same result.

Graph: A graph consists of vertices and edges. Each edge is connected to a source node and a target node. Vertices and edges can be labeled with numbers and symbols.

Graph Production: Similar to productions in general Chomsky grammars, a graph production consists of a left hand side and a right hand side. The left hand side is embedded in a host graph. Then it is removed and in the resulting hole the right hand side of the graph production is inserted. To specify how this right hand side is attached into this hole, how edges are connected to the new nodes, some additional information is necessary. Different approaches exist how to handle this problem.

Graph Rewriting: The application of a graph production to a graph is also called graph rewriting.

Neuron: The smallest processing unit in a neural network.

Neural Networks: Learning systems, designed by analogy with a simplified model of the neural connections in the brain, which can be trained to find nonlinear relationships in data. Several neurons are connected to form the neural networks.

Probabilistic Neural Network: One of the many different kinds of neural networks with the application area to classify input data into different classes.

Rewrite System: A rewrite system consists of a set of configurations and a relation $x \rightarrow y$ denoting that the configuration x follows the configuration y with the help of a rule application.

Termination: A rewrite system terminates if it has no infinite chain.

Weight: Connections between neurons of neural networks have a weight. This weight can be changed during the training of the net.

New Opportunities in Marketing Data Mining

Victor S.Y. Lo
Fidelity Investments, USA

INTRODUCTION

Data mining has been widely applied in many areas over the past two decades. In marketing, many firms collect large amount of customer data to understand their needs and predict their future behavior. This chapter discusses some of the key data mining problems in marketing and provides solutions and research opportunities.

BACKGROUND

Analytics and data mining are becoming more important than ever in business applications, as described by Davenport and Harris (2007) and Baker (2006). Marketing analytics have two major areas: market research and database marketing. The former addresses strategic marketing decisions through survey data analysis and the latter handles campaign decisions through analysis of behavioral and demographic data. Due to the limited sample size of a survey, market research is normally not considered data mining. This chapter will focus on database marketing where data mining is used extensively by large corporations and consulting firms to maximize marketing return on investment.

The simplest tool is RFM where historical purchase recency (R), frequency (F), and monetary (M) value are used for targeting. Other tools include profiling by pre-selected variables to understand customer behavior, segmentation to group customers with similar characteristics, and association rules to explore purchase relationships among products, see Rud (2001); Berry and Linoff(2000). More advanced marketing involves predictive modeling to improve targeting and maximize returns. For examples, marketing-mix analysis has been around for three decades to optimize advertising dollars, see Dekimpe and Hanssens (2000); attrition modeling is used to identify customers at risk of attrition, see Rud (2001); and long-term value is used to prioritize marketing and services, see Peppers and Rogers (1997,1999).

To improve 1:1 marketing campaigns (e.g. direct mails, outbound), response modeling to identify likely responders is now a standard practice in larger corporations. As summarized in Figure 1, a previous campaign provides data on the 'dependent variable' (responded or not), which is merged with individual characteristics. A response model is developed to predict the response rate given the characteristics. The model is then used to score the population to predict response rates for all individuals. Finally, the best list of individuals will be targeted in the next campaign in order to maximize effectiveness and minimize expense.

Response modeling can be applied in the following activities via any marketing channel, see Rud (2001); Berry and Linoff (1997):

1. **Acquisition:** Which prospects are most likely to become customers.
2. **Development:** Which customers are mostly likely to purchase additional products (cross-selling) or add monetary value (up-selling).
3. **Retention:** Which customers are most retainable; this can be relationship or value retention.

MAIN FOCUS

In this chapter, we describe highly important problems that are infrequently mentioned in academic literature but frequently faced by marketing analysts. These problems are embedded in various components of the campaign process, from campaign design to response modeling to campaign optimization, see Figure 2. Each problem will be described in the Problem-Solution-Opportunity format.

CAMPAIGN DESIGN

The design of a marketing campaign is the starting point of a campaign process. It often does not receive enough attention in data mining. A poorly designed campaign

Figure 1. Response modeling process

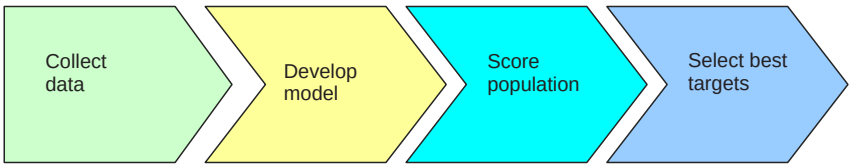
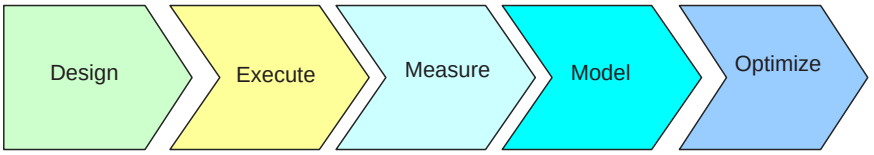


Figure 2. Database marketing campaign process



could make learning infeasible while a scientifically designed one can maximize learning opportunities.

The design process includes activities such as sample size determination for treatment and control groups (both have to be sufficiently large such that measurement and modeling, when required, are feasible), sampling methods (pure or stratified random sampling), and cell design structure (testing various offers and also by age, income, or other variables). We will focus on the latter here.

Problem 1: Classical designs often test one variable at a time. For example, in a cell phone direct mail campaign, we may test a few price levels of the phone. After launching the campaign and the price level that led to the highest revenue is uncovered, another campaign can be launched to test monthly fee and a third campaign will test the mail message, etc. A more efficient way is to structure the cell design such that all these variables are testable in one campaign. Consider an example, a credit card company would like to determine the best combination of treatments for each prospect, and the treatment attributes and attribute levels are summarized in Table 1 (channel is incorporated as an attribute). The number of all possible combinations = $4^4 x 2^2 = 1024$ cells, which is not practical to test.

Solution 1: To reduce the number of cells, a fractional factorial design can be applied (full factorial refers to the design that includes all possible combinations), see Montgomery (1991) and Almquist and Wyner (2001). Two types of fractional factorial are: 1) orthogonal design where all attributes are made orthogonal (uncorrelated) with each other; and 2) optimal design where criterion related to the covariance matrix of parameter estimates is optimized, see Kuhfeld (1997,2004) for the applications of SAS PROC FACTEX and PROC OPTEX in market research (Kuhfeld's market research applications are applicable to database marketing).

For the credit card problem above, an orthogonal fractional factorial design using PROC FACTEX in SAS with estimable main effects, all two-way interaction effects, and quadratic effects on quantitative variables generates a design of 256 cells, which may still be considered large. An optimal design using PROC OPTEX with the same estimable effects generates a design of 37 cells only, see Table 2.

Fractional factorial design has been used in credit card acquisition but is not widely used in other industries for marketing. Two reasons are: (1) lack of experimental design knowledge and experience; (2) business process

Table 1. Example of design attributes and attribute levels

Attribute	Attribute level
APR	5.9% (0), 7.9% (1), 10.9% (2), 12.9% (3)
Credit limit	$2,000 (0), $5,000 (1), $7,000 (2), $10,000 (3)
Color	Platinum (0), Titanium (1), Ruby (2), Diamond (3)
Rebate	None (0), 0.5% (1), 1% (2), 1.5% (3)
Brand	SuperCard (0), AdvantagePlus (1)
Channel	Direct mail (0), Email (1)

Note: numbers in parentheses are coded levels.

requirement—tight coordination with list selection and creative design professionals is required. See Holland (2005) for further applications.

Opportunity 1: Mayer and Sarkissien (2003) proposed using individual characteristics as attributes in an optimal design, where individuals are chosen "optimally." Using both individual characteristics and treatment attributes as design attributes is theoretically compelling. This approach can be evaluated and compared with other techniques such as stratified random sampling. Additionally, if many variables (say, hundreds) are used in the design, constructing an optimal design may be very computationally intensive due to the large design matrix and thus may require a unique optimization technique to solve.

RESPONSE MODELING

Problem 2: As stated in Introduction, response modeling uses data from a previous marketing campaign to identify likely responders given individual characteristics, e.g. Berry and Linoff (1997,2000). Treatment and control groups were set up in the campaign for measurement. The methodology typically uses treatment data to identify characteristics of customers who are likely to respond. In other words, for individual $i \in W$,

$$P(Y_i = 1 | X_i; \text{treatment}) = f(X_i) \quad (1)$$

where W = set of all individuals in the campaign, Y_i = dependent variable (responded or not), and X_i is a set of individual characteristics. For examples, f(.) is a logistic function in logistic regression, Hosmer and

Table 2. An optimal design example (see Table 1 for attribute level definitions)

cell	apr	limit	rebate	color	brand	channel
1	0	0	0	3	0	1
2	0	0	3	3	1	1
3	0	0	3	3	0	0
4	0	0	0	2	1	1
5	0	0	0	2	0	0
6	0	0	0	1	0	1
7	0	0	0	0	1	0
8	0	0	3	0	0	1
9	0	1	3	2	1	0
10	0	3	0	3	1	0
11	0	3	3	3	0	1
12	0	3	3	2	0	1
13	0	3	2	1	1	1
14	0	3	3	1	0	0
15	0	3	0	0	0	1
16	0	3	3	0	1	1
17	1	0	3	1	1	0
18	1	2	0	1	1	0
19	1	3	2	0	0	0
20	2	0	1	0	1	1
21	2	1	3	1	1	1
22	2	3	0	1	0	1
23	3	0	0	3	1	0
24	3	0	3	3	0	1
25	3	0	3	2	1	1
26	3	0	3	2	0	0
27	3	0	0	1	1	1
28	3	0	0	0	0	0
29	3	0	3	0	1	0
30	3	1	0	2	0	1
31	3	1	2	1	0	0
32	3	3	0	3	1	1
33	3	3	0	3	0	0
34	3	3	3	3	1	0
35	3	3	0	2	1	0
36	3	3	0	0	1	0
37	3	3	3	0	0	1

Table 3. Campaign measurement of model effectiveness (response rates)—illustrative

	Treatment (e.g. mail)	Control (e.g. no-mail)	Incremental (treatment minus control)
Model	1%	0.8%	0.2%
Random	0.5%	0.3%	0.2%

Lemeshow (1989); a step function in decision trees, Zhang and Singer (1999); and a nonlinear function in neural network, Haykin (1999).

A problem can arise when we apply model (1) to the next campaign. Depending on the industry and product, the model can select individuals who are likely to respond regardless of the marketing campaign. The results of the new campaign may show equal or similar (treatment minus control) performance in model and random cells, see e.g. Table 3.

Solution 2: The key issue is that fitting model (1) and applying the model to find the 'best' individuals is inappropriate. The correct way is to find those who are most positively influenced by the campaign. It requires us to predict "lift" := $P(Y_i = 1| X_i$; treatment) - $P(Y_i = 1| X_i$; control) for each i and then select those with the highest values of estimated lift. Alternative solutions to this problem are:

1. Fitting two separate treatment and control models: This is relatively straightforward except that estimated lift can be sensitive to statistically insignificant differences in parameters of the treatment and control models;
2. Using dummy variable and interaction effects: Lo (2002) proposed the following independent variables: X_i, T_i, and X_i*T_i where $T_i = 1$ if i is in the treatment group and = 0 otherwise, and modeling the response rate using a logistic regression:

$$P_i = \frac{\exp(\alpha + \beta' X_i + \gamma T_i + \delta' X_i T_i)}{1 + \exp(\alpha + \beta' X_i + \gamma T_i + \delta' X_i T_i)} \qquad (2)$$

where α, β, γ, δ, are parameters to be estimated. In equation (2), α denotes the intercept, β is a vector of parameters measuring the main effects of the independent variables, γ denotes the main treatment effect, and δ measures additional effects of the independent variables due to treatment. To predict the lift:

$$P_i \mid treatment - P_i \mid control = \frac{\exp(\alpha+\gamma+\beta' X_i+\delta' X_i)}{1+\exp(\alpha+\gamma+\beta' X_i+\delta' X_i)} - \frac{\exp(\alpha+\beta' X_i)}{1+\exp(\alpha+\beta' X_i)} \qquad (3)$$

That is, the parameter estimates from model (2) can be used in equation (3) to predict the lift for each i in a new data set. Individuals with high positive predicted lift values will be selected for next campaign. Similar technique of using a dummy variable and interactions can be applied in other supervised learning algorithms such as neural network. See Lo (2002) for a detailed description of the methodology.

3. Decision tree approach: To maximize lift, the only commercial software known is Quadstone that uses a decision tree such that each split at every parent node is processed to maximize the difference between lift (treatment—control response rates) in left and right children nodes, see Radcliffe and Surry (1999) and Radcliffe (2007).

Opportunity 2: (1) Alternative modeling methods for this lift-based problem should exist and readers are encouraged to uncover them. (2) As in typical data mining, the number of potential predictors is large. Variable reduction techniques are available for standard problems (i.e. finding the best subset associated with the dependent variable) but a unique one for the lift-based problem is needed (i.e. finding the best subset associated with the lift).

Problem 3: An extension of Problem 2 is when multiple treatments/channels are present. Data can be obtained from an experimentally designed campaign where multiple treatment attributes were tested, e.g. Solution 1. Question: how to incorporate treatment attributes and individual characteristics in the same response model?

Solution 3: An extension of equation (2) is to incorporate both treatment attributes and individual characteristics in the same lift-based response model:

$$P_i = \frac{\exp(\eta_i)}{1+\exp(\eta_i)}$$

where $\eta_i = \alpha + \beta' X_i + \gamma T_i + \delta' X_i T_i + \lambda' Z_i + \theta' Z_i X_i$,

Z_i *is the vector of treatment attributes,*

and λ *and* θ *are additional parameters* ($Z_i = 0$ *if* $T_i = 0$).

In practice, equation (4) can have many variables including many interaction effects and thus multi-collinearity may be a concern. Developing two separate treatment and control models may be more practical.

Opportunity 3: Similar to Opportunity 2, there should be alternative methods to handle multiple treatment response modeling.

OPTIMIZATION

Marketing optimization refers to delivering the best treatment combination to the right people so as to optimize certain criterion subject to constraints. It is a natural step after campaign design and response modeling, especially when multiple treatment combinations were tested in the previous campaign(s), and response model score for each treatment combination and each individual is available, see Problem 3.

Problem 4: The main issue is that the number of decision variables = number of individuals in the targets (n) times number of treatment (& channel) combinations (m). For example, in an acquisition campaign where the target population has n=30 MM prospect individuals and the number of treatment combinations is m=100. Then the number of decision variables is nxm=3 billions. Consider the following binary integer programming problem, see e.g. Cornuejols and Tutuncu (2007):

$$\begin{aligned}
&\textit{Maximize } \pi = \sum_i \sum_j \pi_{ij} x_{ij} \\
&\textit{subject to}: \\
&\sum_i x_{ij} \le \max.\ \#\textit{individuals receiving treatment combination } j, \\
&\sum_i \sum_j c_{ij} x_{ij} \le \exp\textit{ense budget, plus other relevant constra}\text{int s}, \\
&x_{ij} = 0 \textit{ or } 1, \\
&\textit{where } \pi_{ij} = \textit{inc.value received by sending treatment comb. } j \textit{ to individual } i, \\
&\qquad c_{ij} = \cos t \textit{ of sending treatment comb. } j \textit{ to individual } i.
\end{aligned} \tag{5}$$

In model (5), π_{ij} is linked to the predicted incremental response rate of individual i and treatment combination j. For example, if π_{ij} is the predicted incremental response rate (i.e. lift), the objective function is to maximize the total number of incremental responders; if π_{ij} is incremental revenue, it may be a constant times the incremental response rate. Similar forms of model (5) appear in Storey and Cohen (2002) and Ansari and Mela (2003) except that we are maximizing *incremental* value or responders here. In the literature, similar formulations for *aggregate-level* marketing optimization such as Stapleton et al. (2003) and Larochelle and Sanso (2000) are common but not for *individual-level* optimization.

Solution 4: One way to address such a problem is through heuristics which do not guarantee global optimum. For example, to reduce the size of the problem, one may group the individuals in the target population into clusters and then apply linear programming to solve at the cluster level. This is exactly the first stage outlined in Storey and Cohen (2002). The second stage of their approach is to optimize at the individual level for each of the optimal clusters obtained from the first stage. SAS has implemented this methodology in their new software known as Marketing Optimization. One may also consider MarketSwitch at www.marketswitch.com where the problem is solved with exact solution through a mathematical transformation, see Leach (2001).

Opportunity 4: (1) Simulation and empirical studies can be used to compare existing heuristics to global optimization. (2) General heuristics such as simulated annealing, genetic algorithms, and tabu search can be attempted, e.g. Goldberg (1989) and Michalewicz and Fogel (2002).

Problem 5: Predictive models never produce exact results. Random variation of predicted response rates can be estimated in various ways. For example, in the holdout sample, the modeler can compute the standard deviation of lift by decile. More advanced methods such as bootstrapping can also be applied to assess variability, e.g. Efron and Tibshirani (1998). Then how should the variability be accounted for in optimization?

Solution 5: A simple way is to perform sensitivity analysis of response rates on optimization. The downside is that if the variability is high and/or the number

of treatment combinations is large, the large number of possibilities will make it difficult. An alternative is to solve the stochastic version of the optimization problem, using stochastic programming, see Wallace and Ziemba (2005). Solving such a large stochastic programming problem typically relies on Monte Carlo simulation method and can easily become a large project.

Opportunity 5: The two alternatives in Solution 5 may not be practical for large optimization problems. It is thus an opportunity for researchers to develop a practical solution. One may consider using robust optimization which incorporates the tradeoff between optimal solution and conservatism in linear programming, see Fabozzi et al (2007).

FUTURE TRENDS

1. Experimental design is expected to be more utilized to maximize learning.
2. More marketers are now aware of the lift problem in response modeling (see Solution 2) and alternative methods will be developed to address the problem.
3. Optimization can be performed across campaigns, channels, customer segments, and business initiatives so that a more "global" optimal solution is achieved. This requires not only data and response models but also coordination and cultural shift.

CONCLUSION

Several problems with suggested solutions and opportunities for research have been described. While they are not frequently mentioned in academic literature, they are highly valuable and commonly faced by marketing analysts. The solutions involve multiple areas including data mining, statistics, operations research, database marketing, and marketing science. Researchers and practitioners are encouraged to evaluate current solutions and develop new methodologies.

REFERENCES

Almquist, E. and Wyner G. (2001). Boost your marketing ROI with experimental design. *Harvard Business Review*, Oct, 135-141.

Ansari, A. and Mela, C. (2003). E-customization. *Journal of Marketing Research*, XL, 131-145.

Baker, S. (2006, January 23). Math will Rock Your Word. *BusinessWeek*. Retrieved from http://www.businessweek.com/magazine/content/06_04/b3968001.htm

Berry, M.J.A. and Linoff, G.S. (1997). *Data Mining Techniques: For Marketing, Sales, and Customer Support*, Wiley.

Berry, M.J.A. and Linoff, G.S. (2000). *Mastering Data Mining*, Wiley.

Birge, J.R. and Louveaux, F. (1997). *Introduction to Stochastic Programming*, Springer.

Cornuejols, G. and Tutuncu, R. (2007). *Optimization Methods in Finance*, Cambridge.

Davenport, T.H. and Harris, J. (2007). *Competing on Analytics: The Science of Winning*, Harvard Business School Press.

Dekimpe, M.G. and Hanssens, D.M. (2000). Time series models in marketing: Past, present, and future. *Intern. J. of Research in Marketing*, p.183-193.

Efron, B. and Tibshirani, R.J. (1998). *An Introduction to the Boostrap*, CRC.

Fabozzi, F.J., Kolm, P.N., Pachamanova, D.A., and Focardi, S.M. (2007). *Robust Portfolio Optimization and Management*, Wiley.

Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization & Machine Learning*, Addison-Wesley.

Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation*, Prentice Hall.

Holland, C. (2005). *Breakthrough Business Results with MVT*, Wiley.

Hosmer, D.W. and Lemeshow, S. (1989). *Applied Logistic Regression*, Wiley.

Kuhfeld, W. (1997). Efficient experimental designs using computerized searches. *Sawtooth Software Research Paper Series*.

Kuhfeld, W. (2004). Experimental Design, Efficiency, Coding, and Choice Designs. *SAS Technical Note*, TS-694C.

Larochelle, J. and Sanso, B. (2000). An optimization model for the marketing-mix problem in the banking industry. *INFOR*, 38(4), 390-406.

Leach, P. (2001). Capital One optimizes customer relationships. *1to1 Magazine*, August.

Lo, V.S.Y. (2002). The true-lift model—a novel data mining approach to response modeling in database marketing. *SIGKDD Explorations*, 4(2), 78-86.

Mayer, U.F. and Sarkissien, A. (2003). Experimental Design for Solicitation Campaigns. *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 717-722.

Michalewicz, Z. and Fogel, D.B. (2002). *How to Solve It: Modern Heuristics*, Springer.

Montgomery, D.C. (1991). *Design and Analysis of Experiments*, 3rd edition, Wiley, p.79-80.

Peppers, D. and Rogers, M. (1997). *Enterprise One-to-One*, Doubleday.

Peppers, D. and Rogers, M. (1999) *The One-to-One Fieldbook*, Doubleday.

Radcliffe, N.J. and Surry, P. (1999). Differential response analysis: modeling true response by isolating the effect of a single action. *Proceedings of Credit Scoring and Credit Control VI*, Credit Research Centre, U. of Edinburgh Management School.

Radcliffe, N.J. (2007). Using Control Groups to Target on Predicted Lift. *DMA Analytics Annual Journal*, Spring, p.14-21.

Roberts, M.L. and Berger P.D. (1999). *Direct Marketing Management*, Prentice-Hall.

Rud, O.P. (2001). *Data Mining Cookbook*, Wiley.

Stapleton, D.M., Hanna, J.B., and Markussen, D. (2003). Marketing strategy optimization: Using linear programming to establish an optimal marketing mixture. *American Business Review*, 21(2), 54-62.

Storey, A. and Cohen, M. (2002). Exploiting response models: optimizing cross-sell and up-sell opportunities in banking *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 325-331.

Wallace, S.W. and Ziemba, W.T. (2005). *Applications of Stochastic Programming*, MPS-SIAM.

Zhang, H. and Singer, B. (1999). *Recursive Partitioning in the Health Sciences*, Springer.

KEY TERMS

Campaign Design: The art and science of designing a marketing campaign including cell structure, sampling, and sizing.

Control Group: Individuals who look alike those in the treatment group but are not contacted.

Database Marketing: A branch of marketing that applies database technology and analytics to understand customer behavior and improve effectiveness in marketing campaigns.

Fractional Factorial: A subset of the full factorial design, i.e. subset of all possible combinations.

Marketing Optimization: Delivering the best treatments to the right individuals.

Response Modeling: Predicting response given individual characteristics using data from a previous campaign.

Treatment Group: Individuals who are contacted (e.g. mailed) in a campaign.

Non-Linear Dimensionality Reduction Techniques

Dilip Kumar Pratihar
Indian Institute of Technology, Kharagpur, India

INTRODUCTION

Most of the complex real-world systems involve more than three dimensions and it may be difficult to model these higher dimensional data related to their input-output relationships, mathematically. Moreover, the mathematical modeling may become computationally expensive for the said systems. A human being can visualize only up to three dimensions (3-D). So, any system involving more than 3-D cannot be visualized. To overcome this difficulty, higher dimensional data are generally mapped into either 2-D or 3-D, for visualization and ease of modeling. Dimensionality reduction techniques are nothing but the mapping methods, with the help of which the higher dimensional data can be mapped into the lower dimension after ensuring a reasonable accuracy. It is to be noted that the precision of modeling depends on the said accuracy in mapping. Thus, it is worthy to study the dimensionality reduction techniques.

BACKGROUND

A number of dimensionality reduction techniques are available in the literature; those are classified into two groups, namely linear and non-linear methods (Siedlecki, Seidlecka & Sklansky, 1988; Konig, 2000; Pratihar, Hayashida & Takagi, 2001). In linear methods, each of the lower dimensional components is considered as a linear combination of the higher dimensional components. These methods include principal component analysis (Jolliffe, 1986; Jackson, 1991), projection pursuit mapping (Crawford & Fall, 1990), factor analysis (Mardia, Kent & Bibby, 1995), independent component analysis (Cardoso, 1999), and others. On the other hand, there are some non-linear mapping methods in use, in which the relationships among the lower dimensional and higher dimensional components are non-linear in nature. The non-linear methods are further classified into two sub-groups, namely distance preserving and topology preserving techniques. Distance preserving techniques include Sammon's non-linear mapping (NLM) (Sammon, 1969), VISOR algorithm (Konig, Bulmahn & Glessner, 1994), triangulation method (Konig, 2000), and others, whereas the techniques like self-organizing map (SOM) (Kohonen, 1995), topology preserving mapping of sample sets (Konig, 2000) are known as the topology preserving tools. Two other dimensionality reduction techniques, namely locally linear embedding (LLE) (Roweis & Saul, 2000) and Isomap (Tenenbaum, de Silva & Langford, 2000) proposed in 2000, have also gained the popularity. Dimensionality reduction problem has also been solved using an optimization tool like genetic algorithm (GA) (Raymer, Punch, Goodman, Kuhn & Jain, 2000). In this connection, it is important to mention that more recently, the author of this chapter along with one of his students have proposed a GA-like approach for dimensionality reduction (Dutta & Pratihar, 2006). Another dimensionality reduction technique, namely HyperMap has been proposed by An, Yu, Ratanamahatana & Phoebe Chen (2007), in which the limitation of Euclidean space has been overcome by representing an axis as a line, a plane or a hyperplane. Moreover, an interactive technique has been developed to vary the weights associated with each hyperaxis for ease of visualization. The present chapter deals with some of the non-linear dimensionality reduction techniques (also known as mapping methods). It is important to mention that a particular mapping method may differ from others in terms of accuracy in mapping, visualization capability and computational complexity. An ideal dimensionality reduction technique is one, which can carry out the mapping from a higher dimensional space to a lower dimensional space accurately at a low computational cost and at the same time, can also provide with the widely distributed mapped data suitable for visualization.

MAIN FOCUS

In this section, the principles of some of the non-linear dimensionality reduction techniques have been explained.

Sammon's Nonlinear Mapping (NLM)

It is a distance preserving technique of mapping. Here, the error in mapping from a higher dimensional space to a lower dimensional space/plane is minimized using a gradient descent method (Sammon, 1969).

Let us consider N points in an L-dimensional (L>3) space represented by X_i, where i = 1,2,3,....,N. The aim is to map these N-points from L-dimensional space to 2-D plane or 3-D space. Let us also suppose that the mapped data in 2-D plane or 3-D space are represented by Y_i, where i = 1,2,....,N. The scheme is shown in Figure 1.

The N points in L-dimensional space are indicated as follows:

$$X_1 = \begin{bmatrix} x_{11} \\ x_{12} \\ . \\ . \\ x_{1L} \end{bmatrix}, X_2 = \begin{bmatrix} x_{21} \\ x_{22} \\ . \\ . \\ x_{2L} \end{bmatrix}, \ldots\ldots, X_N = \begin{bmatrix} x_{N1} \\ x_{N2} \\ . \\ . \\ x_{NL} \end{bmatrix}.$$

Similarly, the N-points in 2-D plane or 3-D space can be expressed like the following.

$$Y_1 = \begin{bmatrix} y_{11} \\ . \\ . \\ . \\ y_{1D} \end{bmatrix}, Y_2 = \begin{bmatrix} y_{21} \\ . \\ . \\ . \\ y_{2D} \end{bmatrix}, \ldots\ldots\ldots, Y_N = \begin{bmatrix} y_{N1} \\ . \\ . \\ . \\ y_{ND} \end{bmatrix},$$

where D indicates the dimension of the lower dimensional space.

This technique consists of the following steps:

- Initially, generate N points in 2-D plane at random, corresponding to N points in L-D space.
- Determine the mapping error as follows: Let d_{ij}^* be the Euclidean distance between two points X_i and X_j in L-dimensional space and d_{ij} represents the Euclidean distance between the corresponding two mapped points Y_i and Y_j, in 2-D plane. For an error-free mapping, the following condition has to be satisfied:

  $$d_{ij}^* = d_{ij}.$$

However, the mapping may not be error-free and the mapping error in m^{th} iteration E(m) can be determined mathematically as follows:

$$E(m) = \frac{1}{C} \sum_{i=1}^{N} \sum_{j=1(i<j)}^{N} \frac{[d_{ij}^* - d_{ij}(m)]^2}{d_{ij}^*}$$

where $C = \sum_{i=1}^{N} \sum_{j=1(i<j)}^{N} d_{ij}^*$ and

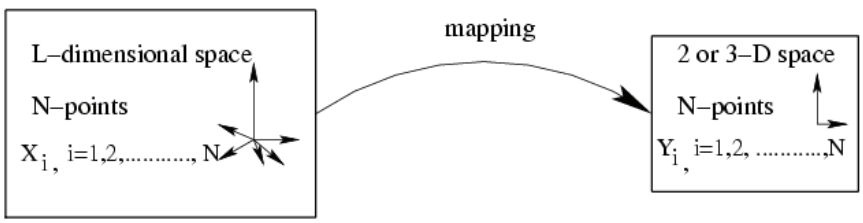
$$d_{ij}(m) = \sqrt{\sum_{k=1}^{D} [y_{ik}(m) - y_{jk}(m)]^2}.$$

- The aforementioned mapping error can be minimized using the steepest descent method, in which the search follows a rule given as:

  $$y_{pq}(m+1) = y_{pq}(m) - (MF)\Delta_{pq}(m),$$

  where $y_{pq}(m)$ and $y_{pq}(m+1)$ represent the $q-th$ dimension of point p in 2-D at $m-th$ and $(m+1)-th$ iterations, respectively, MF is a magic factor representing the step length and it varies in the range of 0.0 to 1.0, and:

Figure 1. Mapping from L-dimensional space to 2-D plane or 3-D space using NLM (Dutta & Pratihar, 2006).



$$\Delta_{pq}(m) = \frac{\partial E(m)/\partial y_{pq}(m)}{\left|\partial^2 E(m)/\partial y_{pq}^2(m)\right|}.$$

It is to be noted that the search direction of the algorithm is indicated by Δ_{pq}. It is also important to mention that the search direction has been chosen so, as the rate of change of a function is found to be the maximum along its gradient direction.

The terms $\partial E/\partial y_{pq}$ and $\partial^2 E/\partial y_{pq}^2$ can be expressed as follows (refer to Box 1.).

In this method, the error in mapping is minimized using the principle of a steepest descent method. As the gradient of a function is its local property, there is a chance of the solutions of this method for being trapped into the local minima.

VISOR Algorithm

It is a mapping technique, which uses a geometrical method for data projection by preserving the distance information (Konig, Bulmahn & Glessner, 1994). Its principle is explained in Figure 2.

Let us suppose that N points of L-dimensional space are to be mapped to a 2-D plane as accurately as possible.

Figure 2 shows the scheme of VISOR algorithm, which consists of the following steps:

- Locate the pivot-vectors - V_1, V_2, V_3 in L-dimensional space, which provide a convex enclosure to the remaining data points. The following approach is adopted for the said purpose:

1. Compute centroid M of all the data points N in L-dimensional space by calculating the dimension-wise average values.
2. Identify the pivot-vector V_1 (that is, a point out of all N points) such that the Euclidean distance between V_1 and M follows the condition given as:

   $d(V_1,M) = \max^{N}_{i=1}(d(v_i,M)$.

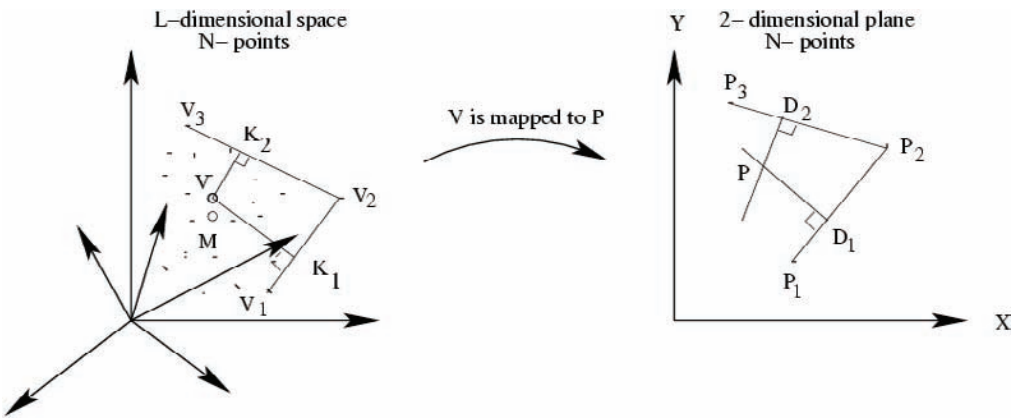
3. Determine the second pivot-vector V_2 (that is, another point out of the remaining $(N-1)$ points) such that the Euclidean distance between V_2 and V_1 is obtained as follows:

Box 1.

$$\frac{\partial E}{\partial y_{pq}} = \frac{-2}{C}\sum_{j=1, j\neq p}^{N}\left[\frac{d^*_{pj} - d_{pj}}{d_{pj} d^*_{pj}}\right](y_{pq} - y_{jq}),$$

$$\frac{\partial^2 E}{\partial y_{pq}^2} = \frac{-2}{C}\sum_{j=1, j\neq p}^{N}\frac{1}{d^*_{pj} d_{pj}}\left[(d^*_{pj} - d_{pj}) - \frac{(y_{pq} - y_{jq})^2}{d_{pj}}\left(1 + \frac{d^*_{pj} - d_{pj}}{d_{pj}}\right)\right]$$

Figure 2. VISOR algorithm (Dutta & Pratihar, 2006)



$d(V_2,V_1) = \max^{N-1}_{i=1}(d(v_i,V_1)$.

4. Locate the pivot-vector V_3 (that is, another point out of the remaining $(N-2)$ points) such that the Euclidean distance between V_3 and V_2 and that between V_3 and V_1 follow the conditions:

   $d(V_3,V_2) = \max^{N-2}_{i=1}(d(v_i,V_2)$ and
   $d(V_3,V_1) = \max^{N-2}_{i=1}(d(v_i,V_1)$.

- Obtain the pivot-vectors - P_1,P_2,P_3 in 2-D plane corresponding to the vectors V_1,V_2,V_3, respectively, by considering their Euclidean distances from the origin of coordinate systems and that between themselves.
- Map the remaining (N-3) points as follows:

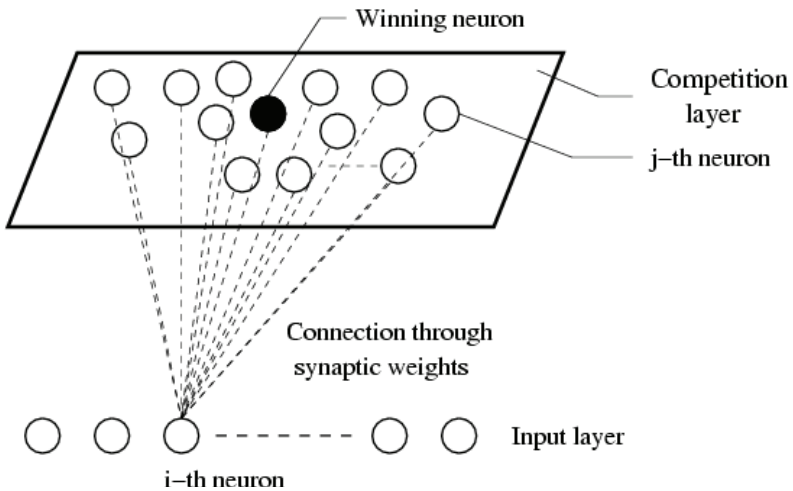
1. Let us suppose that a particular point V of L-dimensional space (refer to Figure 2) is to be mapped to a corresponding point in 2-D plane. The straight lines V_1V_2 and V_2V_3 are considered first and then two perpendicular lines are drawn from the point V to them. Thus, the points K_1 and K_2 are obtained on the lines V_1V_2 and V_2V_3, respectively.
2. In 2-D, locate D_1 on the line P_1P_2 such that D_1 divides the line P_1P_2 in the same proportion as K_1 has divided the line V_1V_2 (in L–dimensional space). Similarly, the point D_2 is obtained on the line P_2P_3.
3. The perpendiculars are drawn at the points D_1 and D_2 to the lines P_1P_2 and P_2P_3, respectively and they intersect at the point P. Thus, the point V in L-dimensional space is mapped to a corresponding point P in 2 – D plane.

As the algorithm uses simple geometrical rules, it is easy to understand and implement.

Self-Organizing Map (SOM)

It is a topology preserving dimensionality reduction technique proposed by Kohonen (Kohonen, 1995). It produces self-organizing feature maps similar to those present in our brain. It works based on unsupervised and competitive learning. Figure 3 shows the schematic diagram of a self-organizing map. It consists of two layers, namely input and competition layers. The input layer contains multivariate data, which are to be mapped to a lower dimensional space. The competition layer contains a number of neurons equal to that of the data points present in the input layer. Each multivariate data present in the input layer is connected to all the neurons of the competition layer through some synaptic weights. The neurons in the competition layer undergo three basic operations, such as competition, cooperation and updating, in stages. In the competition stage, neurons in the competition layer compete among themselves to be identified as a winner, which has the best match with the input unit. In the cooperation stage, a neighborhood surrounding the winning neuron is identified, which contains neurons similar to the winning one. The winning neuron along with its

Figure 3. A schematic diagram representing self-organizing map (Dutta & Pratihar, 2006).



neighbors is updated in the third stage. All these stages are explained as follows.

1. **Competition:** Let us assume that there are *L* - dimensional *N* points (neurons) in the input layer. Thus, a particular input neuron *i* can be expressed as follows:

   $X_i = [x_{i1}, x_{i2}, ..., x_{iL}]^T$, where $i = 1,2,...,N$.

   Let us also consider that the synaptic weight vector between input neuron *i* and neuron j lying in the competition layer is denoted by:

   $W^i_j = [w^i_{j1}, w^i_{j2}, ..., w^i_{jL}]^T$, where $j = 1,2,....,N$.

   The number of neurons lying in the competition layer is assumed to be equal to that of the input layer. To identify the best match for the input data *i*, we need to find the neuron lying in the competition layer, which has the smallest Euclidean distance with it. Let us represent the neuron lying in the competition layer that has the best match with the input vector X_i by n. Thus, Euclidean distance between n and X_i can be expressed as follows:

   $n(X_i)$ = Minimum of $\sqrt{(X_i - W^i_j)^2}$,
   where $j = 1,2,...,N$.

2. **Cooperation:** The winning neuron decides its topological neighborhood (expressed with the help of a neighborhood function) of excited neurons and they will cooperate with each other. Due to this cooperation, their synaptic weights will be updated. For cooperation, the neighborhood function is assumed to follow a Gaussian distribution as:

$$h_{j,n(x_i)}(t) = \exp\left(-\frac{d^2_{j,n(X_i)}}{2\sigma^2_t}\right)$$

   where iteration number $t = 0,1,2,...$ and $d_{j,n(X_i)}$ represents the lateral distance between the winning neuron n and the excited neuron j, σ_t indicates the value of standard deviation at t^{th} iteration, which is expressed as $\sigma_t = \sigma_0 \exp(-(t/\tau))$, where σ_0 is the initial value of standard deviation and τ indicates the pre-defined number of maximum iterations. Thus, the topological neighborhood is assumed to shrink with the number of iterations. It indicates that the algorithm initially starts with a wide neighborhood for cooperation and the size of neighborhood is decreased iteratively. As the iteration proceeds, the algorithm becomes more selective in choosing the neighborhood for cooperation. Thus, the winning neuron and its neighbors help each other to improve their situations.

- **Updating:** The synaptic weights of the winning neuron and all the excited neurons lying in its neighborhood are updated by following the rule:

$$W^i_j(t+1) = W^i_j(t) + \eta(t) h_{j,n(X_i)}(t)[X_i - W^i_j(t)],$$

  where $\eta(t)$ is the learning rate lying between 0.0 and 1.0.
  The aforementioned method is followed to identify a winning neuron lying on the competition layer corresponding to each of the input data points. The winning neurons are then displayed in the lower dimension for visualization. As it is a topology-preserving tool, the neighbourhood information of the higher dimensional data will remain intact in the lower dimension.

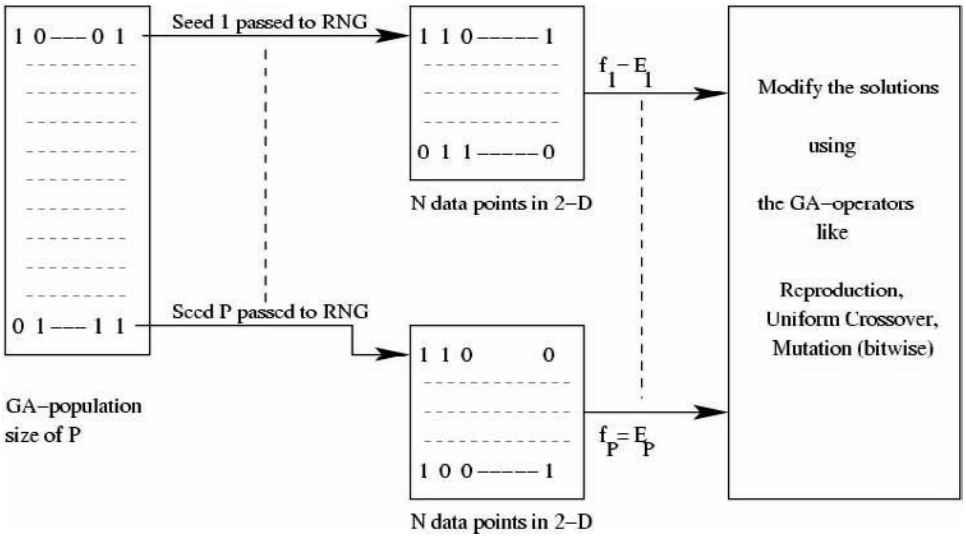
Genetic Algorithm-Like Technique

Genetic algorithm (GA) is a non-traditional tool of optimization, in which different operators of natural genetics have been modeled artificially and the principle of natural selection has been followed (Holland, 1975; Goldberg, 1989). A GA-like technique has been proposed by Dutta & Pratihar, to map data from a higher dimensional space to a lower dimensional space after preserving the distance information (Dutta & Pratihar, 2006).

The technique consists of the following steps (refer to Figure 4):

- Generate a set of N random data points lying on the surface of a higher dimensional function after considering the values of the input variables within their respective ranges. Determine the Euclidean distance values d^*_{ij} s for different combinations of two points - *i* and *j*.

Figure 4. A schematic diagram of the GA-like dimensionality reduction technique (Dutta & Pratihar, 2006)



- Create an initial population of solutions of size *P*, at random. Some bits (consisting of 1 and/or 0) are assigned to represent each solution. The decoded value of each binary string (solution) will be used as a seed for another Random Number Generator (RNG). Thus, the initial population of solutions will contain P seed values. Using a particular seed value, N data points are generated in 2-D lying within the range of the variables, at random. The Euclidean distance between the points *i* and *j* is then calculated, in 2-D. Thus, the GA-like approach carries out its search in a wider space compared to where the conventional GA does.
- Determine the error in mapping E_p (for p^{th} solution) using the expression:

$$E_p = \frac{1}{C}\sum_{i=1}^{N}\sum_{j=1(i<j)}^{N}\frac{[d_{ij}^{*} - d_{ij}]^2}{d_{ij}^{*}}$$

The symbols carry the same meaning as they have in Sammon's NLM.

- The fitness of p^{th} solution of the population is made equal to E_p, that is to be minimized. The fitness values for all the solutions of the population are calculated by following the same procedure.
- Modify the population of solutions using the operators, namely tournament selection, uniform crossover and bit-wise mutation. Tournament selection is a special type of reproduction scheme, in which a number of solutions (depending on the pre-defined tournament size) are taken from the population at random and the best one is selected and kept in the mating pool. Uniform crossover is a modified multi-point crossover, where the participating parents are checked at each bit position with a probability of 0.5, whether there will be a swapping of their bits.
- Continue the process until the termination criterion (say, maximum number of generations or desired accuracy in the solution) is reached. As it is a population-based search and optimization technique, the chance of its solutions for being trapped into the local minima is less.

Comparisons

An attempt has been made to compare the performances of the aforementioned mapping tools on different test functions, in terms of accuracy in mapping, visibility of the mapped data and computational complexity (Dutta & Pratihar, 2006). In this chapter, the results of comparison for a particular test function, say Schaffer's F1 have been shown. Two hundred points are generated at random, which are lying on the surface of the said test function. It is mathematically expressed as:

$$y = 0.5 + \frac{Sin^2\sqrt{\sum_{i=1}^{4} x_i^2} - 0.5}{1.0 + 0.001\left(\sum_{i=1}^{4} x_i^2\right)^2}$$

Figure 5. Results of mapping on Schaffer's F1 function using the approaches – (a) Sammon's NLM, (b) VISOR algorithm, (c) SOM and (d) GA-like approach (Dutta & Pratihar, 2006).

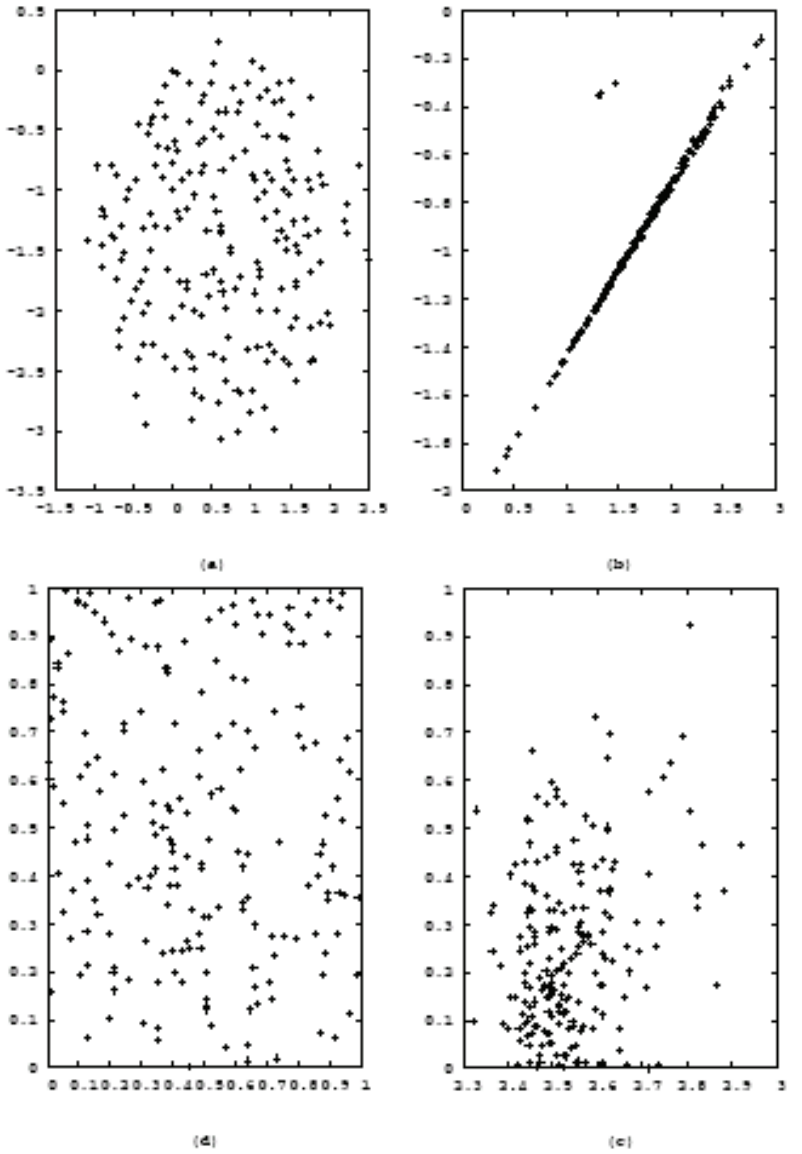


Figure 5 shows the mapped data for the said function in 2-D obtained using the aforementioned four mapping methods. The GA-like approach is found to outperform other approaches in terms of accuracy in mapping, ease of visibility of the mapped data but at the cost of computational complexity. The VISOR algorithm is seen to be the fastest of all but it has been defeated by other algorithms with respect to accuracy in mapping and visibility.

FUTURE TRENDS

An ideal dimensionality reduction technique should be able to map the higher dimensional data to a lower dimension accurately at a low computational cost and at the same time, the mapped data will be well distributed to ensure a better visibility. However, developing an algorithm that can satisfy all the aforementioned requirements in the optimal sense is the main aim of this field of research. Moreover, the mapping techniques could be integrated with other algorithms of data

analysis, such as an optimization tool, a data miner, and others, to have better understanding of their working principle. For example, in order to investigate the topological information of the function to be optimized using a GA, a mapping tool may be clubbed with it for visualization.

CONCLUSION

The present chapter explains the principles of four nonlinear dimensionality reduction techniques. Out of them, the GA-like approach is found to be the best of all in terms of accuracy in mapping and ease of visualization of the mapped data, whereas the VISOR algorithm is seen to be the worst. The GA-like approach and VISOR algorithm are found to be the slowest and fastest of all the aforementioned techniques, respectively. The SOM can also provide with a reasonably well mapping in terms of accuracy and nature of distribution of the mapped data. The performances of the dimensionality reduction techniques are seen to be data dependent.

REFERENCES

An, J., Yu, J. X., Ratanamahatana, C. A., & Phoebe Chen, Y. P. (2007). A dimensionality reduction algorithm and its application for interactive visualization, *Journal of Visual Languages and Computing, 18*, 48-70.

Cardoso, J. F. (1999). ICA Web site. Retrieved from http://www.tsi.enst.fr/icacentral/

Crawford, S., & Fall, T. (1990). Projection pursuit techniques for visualizing high-dimensional data sets. In G. M. Nielson, B. Shriver, & L. J. Rosenblum (Eds.), *Visualization in scientific computing,* (pp. 94-108). Los Alamitos, CA, USA: IEEE Computer Society Press.

Dutta, P., & Pratihar, D., K. (2006). Some studies on mapping methods, *International Journal on Business Intelligence and Data Mining, 1*(3), 347-370.

Goldberg, D. E. (1989). Genetic algorithms in search, optimization, and machine learning. Reading, Mass., USA: Addison-Wesley.

Holland, J. (1975). *Adaptation in natural and artificial systems*. Ann Arbor, USA: The University of Michigan Press.

Jackson, J. E. (1991). *A user's guide to principal components*. New York: John Wiley & Sons.

Jolliffe, I. T. (1986). *Principal component analysis*. Heidelberg, Germany: Springer -Verlag.

Kohonen, T. (1995). *Self-organizing maps*. Heidelberg: Springer-Verlag.

Konig, A., Bulmahn, O., & Glessner, M. (1994). Systematic methods for multivariate data visualization and numerical assessment of class separability and overlap in automated visual industrial quality control. In *Proceedings of 5-th British Machine Vision Conference: Vol. 1*. (pp. 195-204).

Konig, A. (2000). Dimensionality reduction techniques for interactive visualization, exploratory data analysis and classification. In N. R. Pal (Ed.), *Feature analysis, clustering and classification by soft computing,* (pp. 1-37). FLSI Soft Computing Series.

Mardia, K. V., Kent J. T., & Bibby, J. M. (1995). *Probability and mathematical statistics*. New York: Academic Press.

Pratihar, D. K., Hayashida, N., & Takagi, H. (2001). Comparison of mapping methods to Visualize the EC landscape. In *Proceedings of Fifth International Conference on Knowledge-Based Intelligent Information Engineering Systems & Allied Technologies*. Nara, Japan, (pp. 223-227).

Raymer, M. L., Punch, W. F., Goodman, E. D., Kuhn, L.A. & Jain, A. K. (2000). Dimensionality reduction using genetic algorithms. *IEEE Trans. on Evolutionary Computation, 4*(2), 164-171.

Roweis, S. T., & Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science, 290*(1), 2323-2326.

Sammon, J. W. (1969). A nonlinear mapping for data structure analysis. *IEEE Trans. on Computers, C-18*(5), 401-409.

Siedlecki, W., Seidlecka, K., & Sklansky, J. (1988). An overview of mapping techniques for exploratory data analysis. *Pattern Recognition, 21*(5), 411-429.

Tenenbaum, J. B., de Silva, V., & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science,* 290(1), 2319-2323.

KEY TERMS

Dimensionality Reduction: Mapping of higher dimensional data to a lower dimension for ease of visualization and modeling.

Distance Preserving: It is a property, in which the Euclidean distance between two higher dimensional points is kept the same with that between the two corresponding points in lower dimension.

Genetic Algorithm: It is a population-based search and optimization tool, which works based on the mechanics of natural genetics and Darwin's principle of natural selection.

Linear Mapping: It is a mapping, where each of the lower dimensional components of the data set is considered as a linear combination of the higher dimensional components.

Non-linear Mapping: It is a mapping, where each of the lower dimensional components is considered as a non-linear combination of the higher dimensional components.

Self-Organizing Map (SOM): A neural network consisting of input and competition layers that can map the higher dimensional data onto a lower dimension through unsupervised learning.

Steepest Descent Method: It is a gradient-based traditional tool of optimization.

Topology Preserving: It is a property, in which the topological information of a point with its neighborhood in a higher dimensional space will remain unaltered in the lower dimensional space.

Unsupervised Learning: It is a process of learning that takes place through a self-organizing process in the absence of a teacher (that is, the target output).

Visualization: Graphical presentation of data in either 2-D or 3-D for ease of understanding.

A Novel Approach on Negative Association Rules

Ioannis N. Kouris
University of Patras, Greece

INTRODUCTION

Research in association rules mining has initially concentrated in solving the obvious problem of finding positive association rules; that is rules among items that exist in the stored transactions. It was only several years after that the possibility of finding also negative association rules became especially appealing and was investigated. Nevertheless researchers based their assumptions regarding negative association rules on the absence of items from transactions. This assumption though besides being dubious, since it equated the absence of an item with a conflict or negative effect on the rest items, it also brought out a series of computational problems with the amount of possible patterns that had to be examined and analyzed. In this work we give an overview of the works having engaged with the subject until now and present a novel view for the definition of negative influence among items.

BACKGROUND

Association rule mining is still probably the prominent method for knowledge discovery in databases (KDD), among all other methods such as classification, clustering, sequential pattern discovery etc. The discovery of association relationships among items in a huge database has been known to be useful in various sectors such as telecommunications, banking, transport and particularly in retail. Also it has been applied to various data sets such as census data, text documents, transactional data, medical images and lately biological data. In fact any data type that is collected and constitutes a large database of "baskets", each containing multiple "items" can fit this model. The prototypical application of association rules mining and probably until today the most extensively studied and popular one is the analysis of supermarket sales or basket data; hence the problem of finding association rules is often referred to as the "market-basket" problem. In this problem, we are given a set of items and a large collection of transactions which are subsets (baskets) of these items. The objective is to find relationships correlating various items within those baskets. More formally the specific task can be stated as follows:

Let $I=\{i_1, i_2,\ldots,i_n\}$ be a set of items and D be a database organized in multiple transactions T where each transaction $\mathbf{T}\in\mathbf{D}$ is a set of items such that $\mathbf{T}\subseteq\mathbf{I}$. An association rule is an implication of the form $X\rightarrow Y$, where $\mathbf{X, Y}\subseteq\mathbf{I}$ and $\mathbf{X}\cap\mathbf{Y}=\varnothing$, and expresses the possibility that whenever we find a transaction that contains all items in X, then this transaction is likely to also contain all items in Y. Consequently X is called the body of the rule and Y the head. The validity and reliability of association rules is expressed usually by means of support and confidence, succored by other measures such as for example implication, lift etc. An example of such a rule is {cell_phone, hands_free➔case (sup=70%, conf=90%)}, which means that 90% of the customers that buy a cell phone and a hands free device buy a cell phone case as well, whereas 70% of all our customers buy all these three things together.

The task of finding association rules was first formulated by Agrawal, Imielinski and Swami (1993). Since then there have been a series of works trying to improve various aspects such as the efficiency of Apriori based algorithms, implementation of new measures for finding strong – alternative rules, application of association rules framework into other domains etc. (for more details interested reader is referred to the work of Tan, Steinbach & Kumar, 2005).

On the other hand negative association rules have been initially either neglected or considered unimportant. It was not only but after several years since the first introduction of association rules that the problem of negative association rules was brought up. Also despite the fact that the existence and significance of

negative association rules has been recognized for quite some time, comparably a very small percentage of researchers have engaged with it.

MAIN FOCUS

Positive association rules take into consideration only the items that remained and finally appear in the stored transactions. Negative association rules on the other hand, at least as they have been traditionally defined in the literature up to now, take into consideration also the items that do not appear in a transaction. For example a positive association rule would consider all transactions containing pasta and cheese and would generate from them all corresponding association rules (i.e. **pasta → cheese**). An approach dealing with negative association rules on the other hand would also consider rules such as: those that buy pasta buy also cheese but not refreshments (**pasta → cheese** $\wedge \neg$ **refreshment**), or those that buy pasta but not bread buy also cheese (**pasta** $\wedge \neg$ **Bread → cheese**). The measures used for determining the strength of a correlation and for pruning the insignificant ones are again the support and confidence metrics.

A so called generalized negative association rule can include various negated and positive items (i.e. items existing and items absent from transactions) either in its antecedent or in its consequent. An example of such a rule would be: $\mathbf{A} \wedge \neg \mathbf{C} \wedge \neg \mathbf{F} \wedge \mathbf{W} \rightarrow \mathbf{B} \wedge \neg \mathbf{D}$ (i.e. people that buy items A and W but not C and F are likely to buy B but not D). However the obvious insurmountable obstacle created by such a contemplation is the number of possible itemsets that would have to be generated and counted as well as the number of rules created, since apart from the existing items we would have to consider all possible absent items and their combinations. Most approaches thus far have made various assumptions in order to come to approximate solutions, working on subsets of the problem of generalized association rules. For example considering rules where the entire consequent or antecedent could be a conjunction of similar items (i.e. negated or non-negated), considering negative rules only in the infrequent items, considering rules only between two items etc.. Nevertheless the absence of an item from a transaction does not necessarily imply direct negative correlation between items, but could be attributed to various other factors.

Previous Works on Negative Association Rules

To our knowledge Brin, Motwani & Silverstein (1997) were the first that have introduced and addressed the existence of negative relationships between items, by using a chi-squared test for correlations derived from statistics. In the specific work though there were mined negative associations only between two items. Subsequently Savasere, Omiecinski & Navathe (1998) and Yuan, Buckles, Yuan & Zhang (2002) have proposed two similar approaches, which tried to discover strong negative associations depending heavily on domain knowledge and predefined taxonomies. In the same context was the work of Daly & Taniar (2004), who organized items hierarchically following an ancestor – descendant scheme in order to avoid considering all negative items when no additional knowledge would be extracted from lower levels. As an example if customers do not buy refreshments after having bought pasta then we need not examine each and every kind of refreshment. Another work was that of Wu, Zhang & Zhang (2002), which used in addition to the support and confidence measure, a measure for more efficient pruning of the frequent itemsets generated called mininterest. Thiruvady & Webb (2004) proposed an extension of GRD - Generalized Rule Discovery algorithm (Webb, 2000) for mining negative rules. The proposed algorithm did not require the use of any minimum support threshold, but rather the specification of some interestingness measure along with a constraint upon the number of rules that would be finally generated. Finally Antonie & Zaiane (2004) proposed an algorithm for finding generalized association rules, where the entire antecedent or consequent consists of similar items (i.e. only negative or only positive items), thus generating only a subset of negative rules that they refer to as confined negative association rules. In their work they used in addition to the minimum support and minimum confidence measures, the correlation threshold as a third parameter. In the same context can also be considered to some extent works dealing with unexpected patterns (also known as surprising patterns), where the negative items can be thought as such patterns (e.g. Liu, Lu, Feng & Hussain, 1999; Hwang, Ho & Tang, 1999; Hussain, Liu, Suzuki & Lu, 2000; Suzuki & Shimura,

1996; Suzuki, 1997; Padmanabhan & Tuzhilin, 2000; Padmanabhan & Tuzhilin, 1998). Last but not least is the most recent work of Arunasalam, Chawla & Sun (2005) that propose an efficient algorithm for mining both positive and negative patterns in large spatial databases (i.e. databases where objects are characterized by some spatial attributes like location or coordinates).

A NEW APPROACH TO NEGATIVE ITEMSETS

As noted above the equation of the absence of an item from a transaction with a negative influence on the rest items is a highly questionable assumption. The fact that for example beers might never appear together with milk or cereals could possibly mean that these two items present some kind of conflict or negative influence to one another but this is not always the case not even the norm. Practically this kind of behavior could be attributed to various other factors such as (just to name a few for the case of retail data):

- **Complementary products:** Products or items in general that can be consumed or used in place of one another in at least some of their possible uses. Classic examples of substitute goods include margarine instead of butter, or natural gas instead of petroleum.
- **Replacement products:** Products or items where one gradually replaces another one. For example the case where cassettes were replaced by cd's, thus they were never found together in a basket.
- **Mere luck or coincidence** ought to various other factors such as for example due to the placement of products within a store far apart.

Finally the current definition of negative association rules is not only dubious but also creates large computational problems, since the addition of negative items could render the specific task NP – hard.

Our proposal on negative items is based on a far more sound and palpable behavior, similar to that of positive association rules. More specifically instead of considering rules stemming from itemsets where the absence of an item implies a negative relationship we consider itemsets where the conflict between items is recorded in the form of a removal. For example suppose that a customer has already put in his basket pasta, cheese and some refreshment. Normally we would expect the specific customer to keep his basket unchanged until the end or at least to put some more products in it. Nevertheless it could happen that the specific customer now decides to buy beers, but also to remove the refreshment from his basket. Then this is some clear indication that there is some kind of conflict or negative relation between these products, which finally triggered the removal of the refreshment. As one can easily understand this is a completely different contemplation than regarding the absence of an item as an indication of negative relation because in the case of the removal we are dealing with a specific conscious act. This contemplation though raises three main questions that have to be answered. First how are we going to store the actual removal of items? Secondly how are we going to quantify it? And thirdly how are we going to use it?

The answer to the first question is quite trivial. All existing systems and applications do not store the actual removal of an item, but only the cases where it entered and finally remained in a transaction. For example in the case of a retail shop they store for every transaction a unique transaction identifier (noted as TID), the date and sometimes even the exact time of the purchase and finally the unique ids (in the form of bar codes) of the items purchased. Removed items are not taken into consideration in any way, although they contain valuable knowledge that is unfortunately neglected by current approaches and systems. Recording these items though is a simple task that could be implemented even with the existing systems practically with minimum modifications. Of course depending on the application we might have to store not only the order, but the exact time, their physical relevant distance and many other factors.

As far as the quantification of the removal of an item is considered things are a little more complicated. The most obvious and probably naïve approach would be to consider all items preceding the item that was removed as having an equal negative effect on the removed item. More accurate would be to consider the effect that the preceding items have on the removed one as diminishing according to their relevant distance in the transaction (i.e. the item chosen right before the removal having the greatest negative influence). Another possible solution would be to consider as the determinant factor and consequently the only one having a negative

effect on the removed item the one that was purchased last. This is justified also by the fact that the removed item was remaining in the transaction along with all the rest items, and was most probably removed by the choice of the last one.

The removed items could be viewed as separate items from the ones that remained in the transaction (i.e. item A is a different item than item –A) or there could be used a formula for combining the removals and the actual appearances. These are yet a few variations of how one could quantify the removal of an item from transactions or records in general. It remains up to the user and the desired outcome to propose and use a specific variation that better suits specific needs.

Last but not least is the actual use of these patterns. This task is also application specific. For example if we are operating in a retail firm, where the profits generated is the determinant factor, then we would have to examine if the item proposed thus possibly triggering the removal of another one has more net profits than the removed one as well as if it triggers the choice of more other items.

FUTURE TRENDS

The contemplation proposed above has to be followed by the design and implementation of efficient procedures and algorithms for mining negative association rules. Such algorithms might spring from existing ones, although we believe that some completely new solutions would have to be implemented. Another very important parameter neglected in current works in association rules mining is the exact order of items within transactions. The fact that some items are purchased together gains more interest and applicability if we stored and used their exact order of appearances coupled with parameters such as time between purchases, time to complete a purchase etc. This parameter is yet more important if combined with items being removed from transactions. Of course taking into consideration the order of items within transactions adds a lot to complexity, and requires carefully planned approaches.

CONCLUSION

In this chapter we attempted to give a brief but thorough review of all works done thus far in the field of finding negative relationships among items, or had some close relevance to the specific task. Consequently we described the functioning of current approaches, argued about their effectiveness exposing several of their biggest drawbacks that make them less effective. We concluded with a presentation of our view of the problem of finding negative itemsets and relationships among them, possible problems caused by our contemplation as well as some solutions to these problems.

REFERENCES

Agrawal, R., Imielinski, T., & Swami, A. (1993). Mining Association Rules between Sets of Items in Large Databases. In *Proceedings of 1993 ACM-SIGMOD Conference on Management of Data (SIGMOD'93)*, Washington, D.C., 207-216.

Antonie, M.-L., & Zaiane, O., R. (2004). Mining Positive and Negative Association Rules: An Approach for Confined Rules. In *Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'04)*, Springer Verlag LNCS 3202, Pisa, Italy, 27-38.

Arunasalam, B., Chawla, S., & Sun, P. (2005). Striking Two Birds With One Stone: Simultaneous Mining of Positive and Negative Spatial Patterns. In *Proceedings of the Fifth SIAM International Conference on Data Mining (SDM'05)*, Newport Beach, CA, USA, 173-183.

Brin, S., Motwani, R. & Silverstein, C. (1997). Beyond market basket: Generalizing association rules to correlations. In *Proceedings of 1997 ACM-SIGMOD Conference on Management of Data (SIGMOD'97)*, 265-276

Daly, O., & Taniar, D. (2004). Exception rules mining based on negative association rules. In *Computational Science and Its Applications*, 3046, 543-552.

Hussain, F., Liu, H., Suzuki, E., & Lu, H. (2000). Exception rule mining with a relative interestingness measure. In *Proceedings of the 3rd Pacific Asia Conference on Knowledge Discovery and Data Mining (PAKDD'00)*. Springer, Kyoto, Japan, 86-97.

Hwang, S., Ho, S., & Tang, J. (1999). Mining exception instances to facilitate workflow exception handling. In *Proceedings of the 6th International Conference on*

Database Systems for Advanced Applications (DASFAA'99). IEEE Computer Society, Hsinchu, Taiwan, 45-52.

Kleinberg, J., Papadimitriou, C., & Raghavan, P. (1998). A microeconomic view of data mining. *Data Mining and Knowledge Discovery Journal*, 2(4), 311-324.

Liu, H., Lu, H., Feng, L., & Hussain, F. (1999). Efficient search of reliable exceptions. In *Proceedings of the 3rd Pacific Asia Conference on Knowledge Discovery and Data Mining (PAKDD'99)*. Springer-Verlag, Beijing, China, 194-204.

Mannila, H. (1998). Database methods for Data Mining. In *Proceedings of the 4th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'98)*, tutorial.

Padmanabhan, B., & Tuzhilin, A. (2000). Small is beautiful: discovering the minimal set of unexpected patterns. In *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'00)*. ACM, Boston, MA, USA, 54-63.

Padmanabhan, B. & Tuzhilin, A. (1998). A belief-driven method for discovering unexpected patterns. In *Proceedings of the 4th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'98)*. AAAI, Newport Beach, California, USA, 94-100.

Savasere, A., Omiecinski, E., & Navathe, S. B. (1998). Mining for strong negative associations in a large database of customer transactions. In *Proceedings of 14th International Conference on Data Engineering (ICDE'98)*, 494-502

Suzuki, E. (1997). Autonomous discovery of reliable exception rules. In *Proceedings of the 3rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'97)*. AAAI, Newport Beach, California, USA, 259-262.

Suzuki, E., & Shimura, M. (1996). Exceptional knowledge discovery in databases based on information theory. In *Proceedings of the 2nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'96)*. AAAI, Portland, Oregon, USA, 275-278.

Tan, P.-N., Steinbach, M., & Kumar, V. (2005). *Introduction to data mining*. (1st Edition). Bosto: Addison-Wesley Longman Publishing Co.

Teng, W., Hsieh, M., & Chen, M. (2002). On the mining of substitution rules for statistically dependent items. In *Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM '02)*, 442-449

Thiruvady, D., & Webb, G. (2004). Mining negative rules using GRD. In *Proceedings of the 8th Pacific Asia Conference on Knowledge Discovery and Data Mining (PAKDD'04)*. Springer, Sydney, Australia, 161-165.

Yuan, X., Buckles, B., Yuan, Z., & Zhang, J. (2002). Mining negative association rules. In *Proceedings of the 7th International Symposium on Computers and Communications (ISCC'02)*, 623-629

Webb, G. I. (2000). Efficient search for association rules. In *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'00), (pp.* 99-107). Boston: ACM.

Wu, X., Zhang, C., & Zhang, S. (2002). Mining both positive and negative association rules. In *Proceedings of the 19th International Conference on Machine Learning (ICML'02)*, 658-665

KEY TERMS

Association Rules: An implication of the form $X \rightarrow Y$ where X and Y are itemsets. An association rule expresses the possibility that whenever we find a transaction that contains all items in *X*, then this transaction is likely to also contain all items in *Y*.

Confidence: A rule X→Y is said to have confidence c%, if c% of the records (transactions) in a database that contain X contain also Y.

Data Mining: Analysis of data in a database using tools which look for trends or anomalies without knowledge of the meaning of the data. The nontrivial extraction of implicit, previously unknown, and potentially useful information from data. The science of extracting useful information from large data sets or databases.

Itemsets: A collection or a combination of positive or negative items in a database.

Negative Itemsets: Traditionally as negative itemsets were considered those absent from transactions. An alternative contemplation is those itemsets having entered and later have being removed from a transaction.

Positive Association Rules: Rules that value and take into consideration only the items that remain finally in a transaction.

Support: A rule X→Y is said to have support s%, if c% of all the records (transactions) in a database contain both itemsets X and Y.

Offline Signature Recognition

Indrani Chakravarty
Indian Institute of Technology, India

Nilesh Mishra
Indian Institute of Technology, India

Mayank Vatsa
Indian Institute of Technology, India

Richa Singh
Indian Institute of Technology, India

P. Gupta
Indian Institute of Technology, India

INTRODUCTION

The most commonly used protection mechanisms today are based on either what a person possesses (e.g. an ID card) or what the person remembers (like passwords and PIN numbers). However, there is always a risk of passwords being cracked by unauthenticated users and ID cards being stolen, in addition to shortcomings like forgotten passwords and lost ID cards (Huang & Yan, 1997). To avoid such inconveniences, one may opt for the new methodology of Biometrics, which though expensive will be almost infallible as it uses some unique physiological and/or behavioral (Huang & Yan, 1997) characteristics possessed by an individual for identity verification. Examples include signature, iris, face, and fingerprint recognition based systems.

The most widespread and legally accepted biometric among the ones mentioned, especially in the monetary transactions related identity verification areas is carried out through handwritten signatures, which belong to behavioral biometrics (Huang & Yan,1997). This technique, referred to as signature verification, can be classified into two broad categories - online and off-line. While online deals with both static (for example: number of black pixels, length and height of the signature) and dynamic features (such as acceleration and velocity of signing, pen tilt, pressure applied) for verification, the latter extracts and utilizes only the static features (Ramesh and Murty, 1999). Consequently, online is much more efficient in terms of accuracy of detection as well as time than off-line. But, since online methods are quite expensive to implement, and also because many other applications still require the use of off-line verification methods, the latter, though less effective, is still used in many institutions.

BACKGROUND

Starting from banks, signature verification is used in many other financial exchanges, where an organization's main concern is not only to give quality services to its customers, but also to protect their accounts from being illegally manipulated by forgers.

Forgeries can be classified into four types—random, simple, skilled and traced (Ammar, Fukumura & Yoshida, 1988; Drouhard, Sabourin, & Godbout, 1996). Generally online signature verification methods display a higher accuracy rate (closer to 99%) than off-line methods (90-95%) in case of all the forgeries. This is because, in off-line verification methods, the forger has to copy only the shape (Jain & Griess, 2000) of the signature. On the other hand, in case of online verification methods, since the hardware used captures the dynamic features of the signature as well, the forger has to not only copy the shape of the signature, but also the temporal characteristics (pen tilt, pressure applied, velocity of signing etc.) of the person whose signature is to be forged. In addition, he has to simultaneously hide his own inherent style of writing the signature, thus making it extremely difficult to deceive the device in case of online signature verification.

Despite greater accuracy, online signature recognition is not encountered generally in many parts of the world compared to off-line signature recognition, because it cannot be used everywhere, especially where signatures have to be written in ink, e.g. on cheques, where only off-line methods will work. Moreover, it requires some extra and special hardware (e.g. pressure sensitive signature pads in online methods vs. optical scanners in off-line methods), which are not only expensive but also have a fixed and short life span.

MAIN THRUST

In general, all the current off-line signature verification systems can be divided into the following submodules:

- Data Acquisition
- Preprocessing and Noise Removal
- Feature Extraction and Parameter Calculations
- Learning and Verification (or Identification)

Data Acquisition

Off-line signatures do not consider the time related aspects of the signature such as velocity, acceleration and pressure. Therefore, they are often termed as "static" signatures, and are captured from the source (i.e. paper) using a camera or a high resolution scanner, in comparison to online signatures (in which data is captured using a digitizer or an instrumented pen generating signals) (Tappert, Suen, & Wakahara, 1990; Wessels & Omlin, 2000), which do consider the time related or dynamic aspects besides the static features.

Preprocessing

The preprocessing techniques that are generally performed in off-line signature verification methods comprise of noise removal, smoothening, space standardization and normalization, thinning or skeletonization, converting a gray scale image to a binary image, extraction of the high pressure region images, etc.

- **Noise Removal:** Signature images, like any other image may contain noises like extra dots or pixels (Ismail & Gad, 2000), which originally do not belong to the signature, but get included in the image because of possible hardware problems or the presence of background noises like dirt. To recognize the signature correctly, these noise elements have to be removed from the background in order to get the accurate feature matrices in the feature extraction phase. A number of filters have been used as preprocessors (Ismail & Gad, 2000) by researchers to obtain the noise free image. Examples include the mean filter, median filter,

Figure 1. Modular structure of an offline verification system

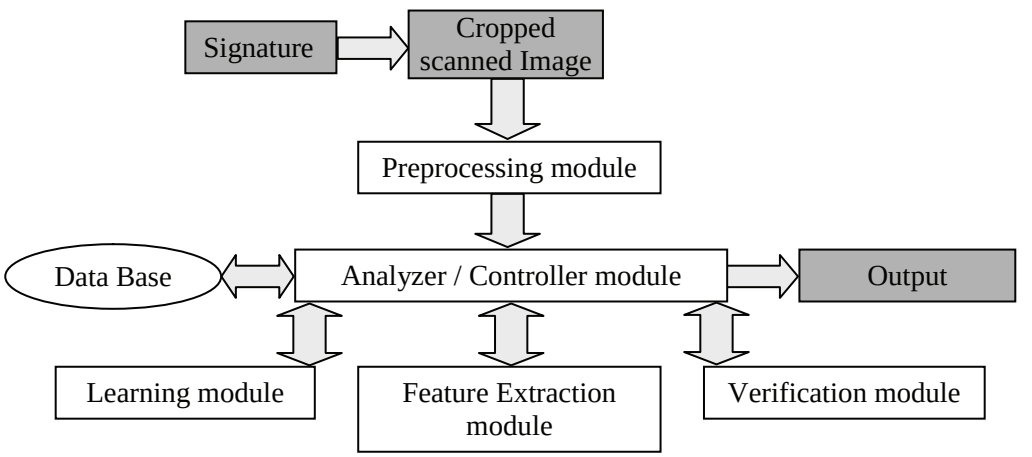
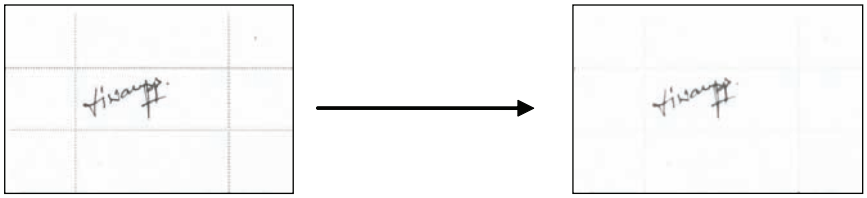


Figure 2. Noise removal using median filter



(a) Gray scale *(b) Noise free*

filter based on merging overlapped run lengths in one rectangle (Ismail & Gad, 2000) etc. Among all the filtering techniques mentioned above, average and median filtering are considered to be standard noise reduction and isolated peak noise removal techniques(Huang & Yan, 1997). However, median filter is preferred more because of its ability to remove noises without blurring the edges of the signature instance unlike the mean filter.

- **Space Standardization and Normalization:** In Space standardization, the distance between the horizontal components of the same signature is standardized, by removing blank columns, so that it does not interfere with the calculation of global and local features of the signature image (Baltzakis & Papamarkos, 2001; Qi & Hunt, 1994). In normalization, the signature image is scaled to a standard size which is the average size of all training samples, keeping the width to height ratio constant (Baltzakis & Papamarkos, 2001; Ismail & Gad, 2000; Ramesh & Murty, 1999).
- **Extracting the Binary Image from Grayscale Image:** Using the Otsu's method a threshold is calculated to obtain a binary version of the grayscale image (Ammar, Fukumura, & Yoshida, 1988; Ismail & Gad, 2000; Qi & Hunt, 1994). The algorithm is as follows:

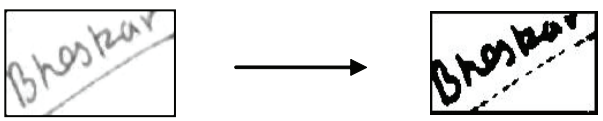
$$S(x,y) = \begin{cases} 1 & R(x,y) > threshold, \\ 0 & R(x,y) < threshold, \end{cases}$$

where S(x,y) is the binary image and R(x,y) is the grayscale image.

- **Smoothening:** The noise-removed image may have small connected components or small gaps, which may need to be filled up. It is done using a binary mask obtained by thresholding followed by morphological operations (which include both erosion and dilation) (Huang & Yan, 1997; Ismail & Gad, 2000; Ramesh & Murty, 1999).
- **Extracting the High Pressure Region Image:** Ammar et al. (Ammar, M., Fukumura, T., & Yoshida Y., 1986) have used the high pressure region as one of the prominent features for detecting skilled forgeries. It is the area of the image where the writer gives special emphasis reflected in terms of higher ink density (more specifically, higher gray level intensities than the threshold chosen). The threshold is obtained as follows:

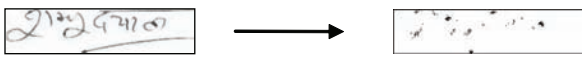
$$Th_{HPR} = I_{min} + 0.75(I_{max} - I_{min}),$$

Figure 3. Converting grayscale image into binary image



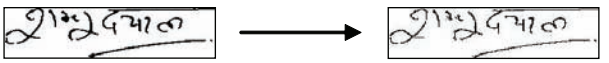
(a) Gray scale *(b) Binary*

Figure 4. Extracting the high pressure region image



(a) Grayscale *(b) High pressure region*

Figure 5. Extraction of thinned image



(a) Binary *(b) Thinned*

where I_{min} and I_{max} are the minimum and maximum grayscale intensity values.

- Thinning or Skeletonization: This process is carried out to obtain a single pixel thick image from the binary signature image. Different researchers have used various algorithms for this purpose (Ammar, Fukumura, & Yoshida, 1988; Baltzakis & Papamarkos, 2001; Huang & Yan, 1997; Ismail & Gad, 2000).

Feature Extraction

Most of the features can be classified in two categories - global and local features.

- **Global Features:** Ismail and Gad (2000) have described global features as characteristics, which identify or describe the signature as a whole. They are less responsive to small distortions and hence are less sensitive to noise as well. Examples include width and height of individual signature components, width to height ratio, total area of black pixels in the binary and high pressure region (HPR) images, horizontal and vertical projections of signature images, baseline, baseline shift, relative position of global baseline and centre of gravity with respect to width of the signature, number of cross and edge points, circularity, central line, corner curve and corner line features, slant, run lengths of each scan of the components of the signature, kurtosis (horizontal and vertical), skewness, relative kurtosis, relative skewness, relative horizontal and vertical projection measures, envelopes, individual stroke segments. (Ammar, Fukumura, & Yoshida, 1990; Bajaj & Chaudhury, 1997; Baltzakis & Papamarkos, 2001, Fang, et al., 2003, Huang & Yan, 1997; Ismail & Gad, 2000; Qi & Hunt, 1994, Ramesh & Murty, 1999; Yacoubi, Bortolozzi, Justino, & Sabourin, 2000; Xiao & Leedham, 2002)
- **Local Features:** Local features are confined to a limited portion of the signature, which is obtained by dividing the signature image into grids or treating each individual signature component as a separate entity (Ismail & Gad, 2000). In contrast to global features, they are responsive to small distortions like dirt, but are not influenced by other regions of the signature. Hence, though extraction of local features requires more computations, they are much more precise. Many of the global features have their local counterparts as well. Examples of local features include width and height of individual signature components, local gradients such as area of black pixels in high pressure region and binary images, horizontal and vertical projections, number of cross and edge points, slant, relative position of baseline, pixel distribution, envelopes etc. of individual grids or components. (Ammar, Fukumura, & Yoshida, 1990; Bajaj & Chaudhury, 1997; Baltzakis & Papamarkos, 2001; Huang & Yan, 1997; Ismail & Gad, 2000; Qi & Hunt, 1994; Ramesh & Murty, 1999; Yacoubi, Bortolozzi, Justino, & Sabourin, 2000)

Learning and Verification

The next stage after feature extraction involves learning. Though learning is not mentioned separately for verification as a separate sub module, it is the most vital part of the verification system in order to verify the authenticity of the signature. Usually five to forty features are passed in as a feature matrix/vector to train the system. In the training phase, 3-3000 signature instances are used. These signature instances include either the genuine or both genuine and forged signatures depending on the method. All the methods extract features in a manner such that the signature cannot be constructed back from them in the reverse order but have sufficient data to capture the features required for verification. These features are stored in various formats depending upon the system. It could be either in the form of feature values, weights of the neural network, conditional probability values of a belief (Bayesian) network or as a covariance matrix (Fang et al., 2003). Later, when a sample signature is to be tested, its feature matrix/vector is calculated, and is passed into the verification sub module of the system, which identifies the signature to be either authentic or unauthentic. So,

Table 1. Summary of some prominent offline papers

Author	Mode of Verification	Database	Feature Extraction	Results
Ammar, Fukumara and Yoshida (1988)	Statistical method-- Euclidian distance and threshold	10 genuine signatures per person from 20 people	Global baseline, upper and lower extensions, slant features, local features (e.g. local slants) and pressure feature	90% verification rate
Huang and Yan (1997)	Neural network Based-- Multilayer perceptron based neural networks trained and used	A total of 3528 signatures (including genuine and forged)	Signature outline, core feature, ink distribution, high pressure region, directional frontiers feature, area of feature pixels in core, outline, high pressure region, directional frontiers, coarse, and in fine ink	99.5 % for random forgery and 90 % for targeted forgeries
Ramesh and Murty (1999)	Genetic Algorithms used for obtaining genetically optimized weights for weighted feature vector	650 signatures, 20 genuine and 23 forged signatures from 15 people	Global geometric features: Aspect ratio, width without blanks, slant angle, vertical center of gravity (COG) etc. Moment Based features: horizontal and vertical projection images, kurtosis measures(horizontal and vertical) Envelope Based: extracting the lower and upper envelopes and Wavelet Based features	90 % under genuine, 98 % under random forgery and 70-80 % under skilled forgery case
Ismail and Gad (2000)	Fuzzy concepts	220 genuine and 110 forged samples	Central line features, corner line features, central circle features, corner curve features, critical points features	95% recognition rate and 98% verification rate
Yacoubi, Bortolozzi, Justino and Sabourin (2000)	Hidden Markov Model	4000 signatures from 100 people (40 people for 1st and 60 for 2nd database	Caliber, proportion, behavior guideline, base behavior, spacing	Average Error(%) :0.48 on 1st database, and 0.88 on the 2nd database
Fang, Leung, Tang, Tse, Kwok, and Wong (2003)	Non-linear dynamic time warping applied to horizontal and vertical projections. Elastic bunch graph matching to individual stroke segments	1320 genuine signatures from 55 authors and 1320 forgeries from 12 authors	Horizontal and Vertical projections for non linear dynamic time warping method. Skeletonized image with approximation of the skeleton by short lines for elastic bunch graph matching algorithm	Best result of 18.1% average error rate (average of FAR and FRR) for first method. The Average error rate was 23.4% in case of the second method

unless the system is trained properly, chances, though less, are that it may recognize an authentic signature to be unauthentic (false rejection), and in certain other cases, recognize the unauthentic signature to be authentic (false acceptance). So, one has to be extremely cautious at this stage of the system. There are various methods for off-line verification systems that are in use today. Some of them include:

- Statistical methods (Ammar, M., Fukumura, T., & Yoshida Y., 1988; Ismail & Gad, 2000)
- Neural network based approach (Bajaj & Chaudhury, 1997; Baltzakis & Papamarkos, 2001; Drouhard, Sabourin, & Godbout, 1996; Huang & Yan, 1997)
- Genetic algorithms for calculating weights (Ramesh & Murty, 1999)
- Hidden Markov Model (HMM) based methods (Yacoubi, Bortolozzi, Justino, & Sabourin, 2000)
- Bayesian networks (Xiao & Leedham, 2002)
- Nonliinear dynamic time warping (in spatial domain) and Elastic bunch graph matching (Fang et al., 2003)

The problem of signature verification is that of dividing a space into two different sets of genuine and forged signatures. Both online and off-line approaches use features to do this, but, the problem with this approach is that even two signatures by the same person may not be the same. The feature set must thus have sufficient interpersonal variability so that we can classify the input signature as genuine or forgery. In addition, it must also have a low intrapersonal variability so that an authentic signature is accepted. Solving this problem using fuzzy sets and neural networks has also been tried. Another problem is that an increase in the dimensionality i.e. using more features does not necessarily minimize(s) the error rate. Thus, one has to be cautious while choosing the appropriate/optimal feature set.

FUTURE TRENDS

Most of the presently used off-line verification methods claim a success rate of more than 95% for random forgeries and above 90% in case of skilled forgeries. Although, a 95% verification rate seems high enough, it can be noticed that, even if the accuracy rate is as high as 99 %, when we scale it to the size of a million, even a 1% error rate turns out to be a significant number. It is therefore necessary to increase this accuracy rate as much as possible.

CONCLUSION

Performance of signature verification systems is measured from their *false rejection rate* (FRR or type I error) (Huang & Yan, 1997) and *false acceptance rate* (FAR or type II error) (Huang & Yan, 1997) curves. Average error rate, which is the mean of the FRR and FAR values, is also used at times. Instead of using the FAR and FRR values, many researchers quote the "100 - average rate" values as the performance result. Values for various approaches have been mentioned in the table above. It is very difficult to compare the values of these error rates, as there is no standard database either for off-line or for online signature verification methods.

Although online verification methods are gaining popularity day by day because of higher accuracy rates, off-line signature verification methods are still considered to be indispensable, since they are easy to use and have a wide range of applicability. Efforts must thus be made to improve its efficiency to as close as that of online verification methods.

REFERENCES

Ammar, M., Fukumura, T., & Yoshida Y. (1986). A new effective approach for off-line verification of signature by using pressure features. *Proceedings 8th International Conference on Pattern Recognition, ICPR'86* (pp. 566-569), Paris.

Ammar, M., Fukumura, T., & Yoshida, Y. (1988). Off-line preprocessing and verification of signatures. *International Journal of Pattern Recognition and Artificial Intelligence 2*(4), 589-602.

Ammar, M., Fukumura, T., & Yoshida, Y. (1990). Structural description and classification of signature images. *Pattern Recognition 23*(7), 697-710.

Bajaj, R., & Chaudhury, S. (1997). Signature verification using multiple neural classifiers. *Pattern Recognition 30*(1), l-7.

Baltzakis, H., & Papamarkos, N. (2001). A new signature verification technique based on a two-stage neural network classifier. *Engineering Applications of Artificial Intelligence*, *14*, 95-103.

Drouhard, J. P., Sabourin, R., & Godbout, M. (1996). A neural network approach to off-line signature verification using directional pdf. *Pattern Recognition 29*(3), 415-424.

Fang, B., Leung, C. H., Tang, Y. Y., Tse, K. W., Kwok, P. C. K., & Wong, Y. K. (2003). Off-line signature verification by tracking of feature and stroke positions. *Pattern Recognition 36*, 91-101.

Huang, K., & Yan, H. (1997). Off-line signature verification based on geometric feature extraction and neural network classification. *Pattern Recognition, 30*(1), 9-17.

Ismail, M. A., & Gad, S. (2000). Off-line Arabic signature recognition and verification. *Pattern Recognition, 33*, 1727-1740.

Jain, A. K. & Griess, F. D., (2000). *Online Signature Verification*. Project Report, Department of Computer Science and Engineering, Michigan State University, USA.

Qi, Y., & Hunt, B. R. (1994). Signature verification using global and grid features. *Pattern Recognition, 27*(12), 1621-1629.

Ramesh, V.E., & Murty, M. N. (1999). Off-line signature verification using genetically optimized weighted features. *Pattern Recognition, 32*(7), 217-233.

Tappert, C. C., Suen, C. Y., & Wakahara, T. (1990). The State of the Art in Onliine Handwriting Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 12*(8).

Wessels, T., & Omlin, C. W, (2000). A Hybrid approach for Signature Verification. *International Joint Conference on Neural Networks*.

Yacoubi, A. El., Bortolozzi, F., Justino, E. J. R., & Sabourin, R. (2000). An off-line signature verification system using HMM and graphometric features. *4th IAPR International Workshop on Document Analysis Systems* (pp. 211-222).

Xiao, X., & Leedham, G. (2002). Signature verification using a modified Bayesian network. *Pattern Recognition, 35*, 983-995.

KEY TERMS

O

Area of Black Pixels: It is the total number of black pixels in the binary image.

Biometric Authentication: The identification of individuals using their physiological and behavioral characteristics.

Centre of Gravity: The centre of gravity of the image is calculated as per the following equation:

$$X = \sum_{y=1}^{m} (y.P_h[y]) / \sum_{y=1}^{m} P_h[y], \text{ and}$$

$$Y = \sum_{x=1}^{n} (x.P_v[x]) / \sum_{x=1}^{n} P_v[x],$$

where X, Y are the x and y coordinates of the centre of gravity of the image, and P_h and P_v are the horizontal and vertical projections respectively.

Cross and Edge Points: A cross point is an image pixel in the thinned image having at least three eight neighbors, while an edge point has just one eight neighbor in the thinned image.

Envelopes: Envelopes are calculated as upper and lower envelopes above and below the global baseline. These are the connected pixels which form the external points of the signature obtained by sampling the signature at regular intervals.

Global Baseline: It is the median value of pixel distribution along the vertical projection.

Horizontal and Vertical Projections: Horizontal projection is the projection of the binary image along the horizontal axis. Similarly, vertical projection is the projection of the binary image along the vertical axis. They can be calculated as follows:

$$P_h[y] = \sum_{x=1}^{n} \text{black.pixel}(x, y),$$

$$P_v[x] = \sum_{y=1}^{m} \text{black.pixel}(x, y),$$

where m = width of the image and n = height of the image.

Moment Measures: Skewness and kurtosis are moment measures calculated using the horizontal and vertical projections and co-ordinates of centre of gravity of the signature.

Slant: Either it is defined as the angle at which the image has maximum horizontal projection value on rotation or it is calculated using the total number of positive, negative, horizontally or vertically slanted pixels.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 870-875, copyright 2005 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

OLAP Visualization: Models, Issues, and Techniques

Alfredo Cuzzocrea
University of Calabria, Italy

Svetlana Mansmann
University of Konstanz, Germany

INTRODUCTION

The problem of efficiently *visualizing multidimensional data sets* produced by scientific and statistical tasks/processes is becoming increasingly challenging, and is attracting the attention of a wide multidisciplinary community of researchers and practitioners. Basically, this problem consists in visualizing multidimensional data sets by capturing the *dimensionality* of data, which is the most difficult aspect to be considered. Human analysts interacting with high-dimensional data often experience disorientation and cognitive overload. Analysis of high- dimensional data is a challenge encountered in a wide set of real-life applications such as (*i*) biological databases storing massive gene and protein data sets, (*ii*) real-time monitoring systems accumulating data sets produced by multiple, multi-rate streaming sources, (*iii*) advanced *Business Intelligence* (BI) systems collecting business data for decision making purposes etc.

Traditional DBMS front-end tools, which are usually tuple-bag-oriented, are completely inadequate to fulfill the requirements posed by an interactive exploration of high-dimensional data sets due to two major reasons: (*i*) DBMS implement the OLTP paradigm, which is optimized for transaction processing and deliberately neglects the dimensionality of data; (*ii*) DBMS operators are very poor and offer nothing beyond the capability of conventional SQL statements, what makes such tools very inefficient with respect to the goal of visualizing and, above all, interacting with multidimensional data sets embedding a large number of dimensions.

Despite the above-highlighted practical relevance of the problem of visualizing multidimensional data sets, the literature in this field is rather scarce, due to the fact that, for many years, this problem has been of relevance for life science research communities only, and interaction of the latter with the computer science research community has been insufficient. Following the enormous growth of scientific disciplines like *Bio-Informatics*, this problem has then become a fundamental field in the computer science academic as well as industrial research. At the same time, a number of proposals dealing with the multidimensional data visualization problem appeared in literature, with the amenity of stimulating novel and exciting application fields such as the visualization of *Data Mining* results generated by challenging techniques like clustering and association rule discovery.

The above-mentioned issues are meant to facilitate understanding of the high relevance and attractiveness of the problem of visualizing multidimensional data sets at present and in the future, with challenging research findings accompanied by significant spin-offs in the *Information Technology* (IT) industrial field.

A possible solution to tackle this problem is represented by well-known OLAP techniques (Codd et al., 1993; Chaudhuri & Dayal, 1997; Gray et al., 1997), focused on obtaining very efficient representations of multidimensional data sets, called *data cubes*, thus leading to the research field which is known in literature under the terms *OLAP Visualization* and *Visual OLAP*, which, in the remaining part of the article, are used interchangeably.

Starting from these considerations, in this article we provide an overview of OLAP visualization techniques with a comparative analysis of their advantages and disadvantages. The outcome and the main contribution of this article are a comprehensive survey of the relevant state-of-the-art literature, and a specification of guidelines for future research in this field.

BACKGROUND

Formally, given a relational data source $\mathcal{R}$, a data cube $\mathcal{L}$ defined on top of R is a tuple $\mathcal{L} = \langle C, J, \mathcal{H}, \mathcal{M} \rangle$, such that: (*i*) C is the data domain of $\mathcal{L}$ containing (OLAP) *data cells* storing *SQL aggregations*, such as those based on SUM, COUNT, AVG etc, computed over tuples in $\mathcal{R}$; (*ii*) J is the set of *functional attributes* (of $\mathcal{R}$) with respect to which $\mathcal{L}$ is defined, also called *dimensions* of $\mathcal{L}$; (*iii*) $\mathcal{H}$ is the set of *hierarchies* related to dimensions of $\mathcal{L}$; (*iv*) $\mathcal{M}$ is the set of *attributes of interest* (of $\mathcal{R}$) for the underlying OLAP analysis, also called *measures* of $\mathcal{L}$. OLAP data cubes can thus be used to effectively visualize multidimensional data sets and also support interactive exploration of such data sets using a wide set of operators (Han & Kamber, 2000), among which we recall: (*i*) *drill-down*, which descends in a dimension hierarchy of the cube by increasing the level of detail of the measure (and decreasing its level of abstraction); (*ii*) *roll-up*, which is a reverse of drill-down used to aggregate the measure to a coarser level of detail (and a finer level of abstraction); (*iii*) *pivot*, which rotates the dimensions of the cube, thus inducting data re-aggregation. Apart the visualization amenities, OLAP also offers very efficient solutions to the related problem of *representing multidimensional data sets* by means of a wide set of alternatives (Han & Kamber, 2000) according to which data cubes are stored in mass memory: (*i*) ROLAP (*Relational OLAP*), which makes use of the storage support provided by conventional RDBMS (i.e., relational tables); (*ii*) MOLAP (*Multidimensional OLAP*), which employs multidimensional arrays equipped with highly-efficient indexing data structures; (*iii*) HOLAP (*Hybrid OLAP*), which combines the two previous alternatives via storing portions of the cube on a relational support, and other portions on an array-oriented support (depending on various parameters such as the query-workload of the cube). Without further details, it is worth noticing that the efficiency of the data representation has a great impact on the effectiveness of data visualization and exploration activities.

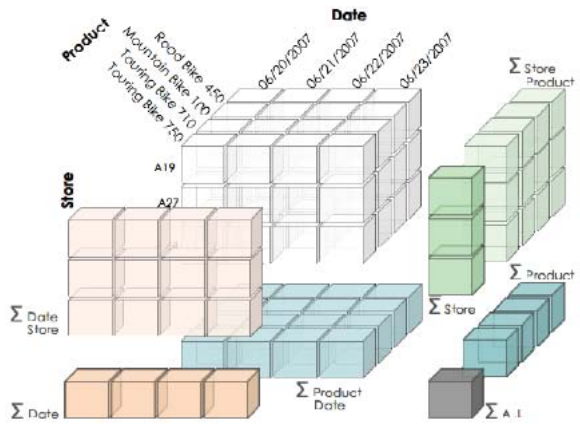
Visual OLAP results from the convergence of BI techniques and the achievements in the scientific areas of *Information Visualization* and *Visual Analytics*. Traditional OLAP front-end tools, designed to support reporting and analysis routines primarily, use visualization merely for expressive presentation of the data. In the Visual OLAP approach, however, visualization plays the key role as the method of *interactive query-driven analysis*. A more comprehensive analysis of such a kind includes a variety of tasks such as: examining the data from multiple perspectives, extracting useful information, verifying hypotheses, recognizing trends, revealing patterns, gaining insights, and discovering new knowledge from arbitrarily large and/or complex volumes of multidimensional data. In addition to conventional operations of analytical processing, such as drill-down, roll-up, slice-and-dice, pivoting, and ranking, Visual OLAP supports further interactive data manipulation techniques, such as zooming and panning, filtering, brushing, collapsing etc.

OLAP VISUALIZATION: A SURVEY

First proposals on using visualization for exploring large data sets were not tailored towards OLAP applications, but addressed the generic problem of visual querying of large data sets stored in a database. Early experiences related to multidimensional data visualization can be found in real-life application scenarios, such as those proposed in (Gebhardt et al., 1997), where an intelligent visual interface to multidimensional databases is proposed, as well as in theoretical foundations, such as those stated in (Inselberg, 2001), which discusses and refines general guidelines on the problem of efficiently visualizing and interacting with high-dimensional data. Keim and Kriegel (1994) propose *VisDB*, a visualization system based on an innovative query paradigm. In *VisDB*, users are prompted to specify an initial query. Thereafter, guided by a visual feedback, they dynamically adjust the query, e.g. by using sliders for specifying range predicates on singleton or multiple attributes. Retrieved records are mapped to the pixels of the rectangular display area, colored according to their degree of relevance for the specified set of selection predicates, and positioned according to a grouping or ordering directive.

A traditional interface for analyzing OLAP data is a *pivot table*, or *cross-tab*, which is a multidimensional spreadsheet produced by specifying one or more measures of interest and selecting dimensions to serve as vertical (and, optionally, horizontal) axes for summarizing the measures. The power of this presentation technique comes from its ability in summarizing detailed data along various dimensions, and arranging aggregates computed at different granularity levels into a single

Figure 1. A three-dimensional data cube £ (left) and the pivot table computed on top of £ (right)



		Measures							
		Quantity				Amount			
Dimensions		Store							
Product	Date	A19	A27	A34	Total Store	A19	A27	A34	Total Store
– Road Bike 450	06/20/2007	2	7	4	13	498	1743	996	3237
	06/21/2007	9	12	10	31	2241	2988	2490	7719
	06/22/2007	3		7	10	747		1743	2490
	06/23/2007	5	1	9	15	1245	249	2241	3735
Total Road Bike 450		19	20	30	69	4731	4980	7470	17181
– Mountain Bike 100	06/20/2007	8	10	3	21	6392	7990	2397	16779
	06/21/2007	5	11	4	20	3995	8789	3196	15980
	06/22/2007	9	7		16	7191	5593		12784
	06/23/2007	6		4	10	4794		3196	7990
Total Mountain Bike 100		28	28	11	67	22372	22372	8789	53533
– Touring Bike 710	06/20/2007	5		9	14	2995		5391	8386
	06/21/2007	7	2	12	21	4193	1198	7188	12579
	06/22/2007	4		13	17	2396		7787	10183
	06/23/2007		2	8	10		1198	4792	5990
Total Touring Bike 710		16	4	42	62	9584	2396	25158	37138
+ Total Touring Bike 750		19	12	15	46	10621	6708	8385	25714
Total Product		82	64	98	244	47308	83765	49802	133566

view preserving the "part-of" relationships between the aggregates themselves. Figure 1 exemplifies the idea of "unfolding" a three-dimensional data cube (left side) into a pivot table (right side), with cells of the same granularity marked with matching background color in both representations. However, pivot tables are inefficient for solving non-trivial analytical tasks, such as recognizing patterns, discovering trends, identifying outliers etc (Lee & Ong, 1995; Eick, 2000; Hanrahan et al., 2007). Despite this weakness point, pivot tables still maintain the power of any visualization technique, i.e. saving time and reducing errors in analytical reasoning via utilizing the phenomenal abilities of the human vision system in pattern recognition (Hanrahan et al., 2007).

OLAP interfaces of the current state-of-the-art enhance the pivot table view via providing a set of popular business visualization techniques, such as bar-charts, pie-charts, and time series, as well as more sophisticated visualization layouts such as scatter plots, maps, tree-maps, cartograms, matrices, grids etc, and vendors' proprietary visualizations (e.g., decomposition trees and fractal maps). Some tools go beyond mere visual presentation of data purposes and propose sophisticated approaches inspired by the findings in Information Visualization research. Prominent examples of advanced visual systems are *Advizor* (Eick, 2000) and *Tableau* (Hanrahan et al., 2007). *Advizor* implements a technique that organizes data into three perspectives. A perspective is a set of linked visual components displayed together on the same screen. Each perspective focuses on a particular type of analytical task, such as (*i*) single measure view using a 3D multi-scope layout, (*ii*) multiple measures arranged into a scatter plot, and (*iii*) anchored measures presented using techniques from multidimensional visualization (box plots, parallel coordinates etc). *Tableau* is a commercialized successor of *Polaris*, a visual tool for multidimensional analysis developed by Stanford University (Stolte et al., 2002). *Polaris* inherits the basic idea of the classical pivot table interface that maps aggregates into a grid defined by dimension categories assigned to grid rows and columns. However, *Polaris* uses embedded graphical marks rather than textual numbers in table cells. Types of supported graphics are arranged into a taxonomy, comprising rectangle, circle, glyph, text, Gantt bar, line, polygon, and image layouts.

Back to basic problems, (Russom, 2000) summarizes trends in business visualization software as a progression from rudimentary data visualization to advanced forms, and proposes distinguishing three life-cycle stages of visualization techniques, such as maturing, evolving, and emerging. Within this classification, Visual OLAP clearly fits into the emerging techniques for advanced interaction and visual querying. In the spirit of Visual OLAP, ineffective data presentation is not the only deficiency of conventional OLAP tools. Further problems are cumbersome usability and poor exploratory functionality. Visual OLAP addresses those problems via developing fundamentally new ways of interacting with multidimensional aggregates. A new quality of visual analysis is achieved via unlocking the synergy between the OLAP technology, Information Visualization, and Visual Analytics.

The task of selecting a proper visualization technique for solving a particular problem is by far not trivial as various *visual representations* (also called *metaphors*) may be not only task-dependent, but also domain-de-

pendent. Successful Visual OLAP frameworks need to be based on a comprehensive taxonomy of domains, tasks, and visualizations. The problem of assisting analysts in identifying an appropriate visualization technique for a specific task is an unsolved issue in state-of-the-art OLAP tools. Typically, a user has to find an appropriate solution manually via experimenting with different layout options. To support a large set of diverse visualization techniques and enable dynamic switching from one technique to another, an abstraction layer has to be defined in order to specify the relationships between data and their visual presentation.

Following this approach, the *Tape* model, proposed by Gebhardt et al. (1998), suggests to represent and visualize multidimensional data domains using the metaphors of *tapes* and *tracks*, enhanced with the possibility of defining *hierarchical structures* within a tape.

Maniatis et al. (2003a; 2003b) propose an abstraction layer solution, called *Cube Presentation Model* (CPM), which distinguishes between two layers: a (*i*) *logical layer*, which deals with data modeling and retrieval, and a (*ii*) *presentation layer*, which provides a generic model for representing the data (normally, on a 2D screen). Entities of the presentation layer include points, axes, multi-cubes, slices, tapes, cross-joins, and content functions. Authors demonstrate how CPM constructs can be mapped onto advanced visual layouts at the example of *Table Lens*, a technique based on a cross-tabular paradigm with support for multiple zoomable windows of focus.

A common approach to visualization in OLAP application relies on a set of templates, wizards, widgets, and a selection of visual formats. Hanrahan et al., (2007) argue however that an open set of requirements cannot be addressed by a limited set of techniques, and choose a fundamentally different approach for their visual analysis tool *Tableau*. This novelty is represented by *VizQL*, a *declarative visual query language*. *VizQL* offers high expressiveness via allowing users to create their own visual presentation by means of combining various visual components. Figure 2 illustrates the visualization approach of *Tableau* via showing just a small subset of sophisticated visual presentations created by means of simple *VizQL* statements not relying on any pre-defined template layout.

Designers of *Tableau* deliberately restrict the set of supported visualizations to the popular and proven ones, such as tables, charts, maps, and time series, as doubting general utility of exotic visual metaphors (Hanrahan et al., 2007). Thereby, *Tableau* approach is constrained to generating grids of visual presentations of uniform granularity and limited dimensionality. Other researchers suggest that Visual OLAP should be enriched by extending basic charting techniques or by employing novel and less-known visualization techniques to take full advantage from multidimensional and hierarchical properties of data (Tegarden, 1999;

Figure 2. VizQL at work (Used by permission of Tableau Software, Inc.)

Lee & Ong, 1995; Techapichetvanich & Datta, 2005; Sifer, 2003). Tegarden (1999) formulates the general requirements of *Business Information Visualization* and gives an overview of advanced visual metaphors for multivariate data, such as *Kiviat Diagrams* and *Parallel Coordinates* for visualizing data sets of high dimensionality, as well as 3D techniques, such as *3D scatter-grams*, *3D line graphs*, *floors* and *walls*, and *3D map-based bar-charts*.

An alternative proposal is represented by the DIVE-ON (*Data mining in an Immersed Visual Environment Over a Network*) system, proposed by Ammoura et al. (2001). The main idea of DIVE-ON is furnishing an immersive visual environment where distributed multidimensional data sources are consolidated and presented to users that can interact with such sources by "walking" or "flying" towards them. Thereby, DIVE-ON makes an intelligent usage of the natural human capability of interacting with spatial objects, thus sensitively enhancing the knowledge fruition phase. In its core layer, DIVE-ON exploits the OLAP technology in order to efficiently support the multidimensionality of data. All considering, we can claim that DIVE-ON is one of the most unique experiences in the OLAP visualization research field, with some characteristics that slightly resemble visual entertainment systems.

Another branch of visualization research for OLAP concentrates on developing multi-scale visualization techniques capable of presenting data at different levels of aggregation. Stolte et al. (2003) describe their implementation of multi-scale visualization within the framework of the *Polaris* system. The underlying visual abstraction is that of a zoom graph that supports multiple zooming paths, where zooming actions may be tied to dimensional axes or triggered by different kinds of interaction. Lee and Ong (1995) propose a multidimensional visualization technique that adopts and modifies the *Parallel Coordinates* method for knowledge discovery in OLAP. The main advantage of this technique is its scalability to virtually any number of dimensions. Each dimension is represented by a vertical axis and aggregates are aligned along each axis in form of a bar-chart. The other side of the axis may be used for generating a bar-chart at a higher level of detail. Polygon lines adopted from the original *Parallel Coordinates* technique are used to indicate relationships among aggregates computed along various dimensions (a relationship exists if the underlying sets of fact entries in both aggregates overlap).

Mansmann and Scholl (2007) concentrate on the problem of losing the aggregates computed at preceding query steps while changing the level of detail, and propose using hierarchical layouts to capture the results of multiple decompositions within the same display. Authors introduce a class of multi-scale visual metaphors called *Enhanced Decomposition Tree*. Levels of the visual hierarchy are created via decomposing the aggregates along a specified dimension, and nodes contain the resulting sub-aggregates arranged into an embedded visualization (e.g., a bar-chart). Various hierarchical layouts and embedded chart techniques are considered to account for different analysis tasks.

Sifer (2003) presents a multi-scale visualization technique for OLAP based on coordinated views of dimension hierarchies. Each dimension hierarchy with qualifying fact entries attached as bottom-level nodes is presented using a space-filling nested tree layout. Drilling-down and rolling-up is performed implicitly via zooming within each dimension view. Filtering is realized via (de-)selecting values of interest at any level of dimension hierarchies, resulting either in highlighting the qualifying fact entries in all dimension views (*global context coordination*) or in eliminating the disqualified entries from the display (*result only coordination*).

A similar interactive visualization technique, called the *Hierarchical Dynamic Dimensional Visualization* (HDDV), is proposed in (Techapichetvanich & Datta, 2005). Dimension hierarchies are shown as hierarchically aligned bar-sticks. A bar-stick is partitioned into rectangles that represent portions of the aggregated measure value associated with the respective member of the dimension. Color intensity is used to mark the density of the number of records satisfying a specified range condition. Unlike in (Sifer, 2003), dimension level bars are not explicitly linked to each other, allowing to split the same aggregate along multiple dimensions and, thus, to preserve the execution order of the dis-aggregation task. A technique for finding appropriate representation of multidimensional aggregates, proposed by Choong et al. (2003), may help to improve the analytical quality of any visualization. This technique addresses the problem of ordering aggregates along dimensional axes. By default, the ordering of the measure values is imposed by the lexical ordering of values within dimensions. To make patterns more obvious, the user has to rearrange the ordering manually. The proposed algorithm automates the ordering of measures in a

representation as to best reveal patterns (e.g., trends and similarity) that may be observed in a data set.

More recently, Cuzzocrea et al. (2006; 2007) propose an innovative framework for efficiently supporting OLAP visualization of multidimensional data cubes. This framework has a wide range of applicability in a number of real-life applications, from the visualization of spatio-temporal data (e.g., mobile data) to that of scientific and statistical data. Based on meaningfully handling OLAP hierarchies of the target data cube $\mathcal{L}$, the novelty of the proposed framework consists in computing a *semantics-based partition* of $\mathcal{L}$ that groups OLAP data cells semantically related, thus originating the so-called *semantics-aware buckets*. Thereafter, the resulting partitioned representation is further compressed by means of highly-efficient quad-tree based data structures, what makes the relevant assumption that compressing massive data cubes is a way for efficiently visualizing these data structures. This compressed representation finally originates a novel multidimensional histogram, called *Hierarchy-driven Indexed Quad-Tree Summary* (H-IQTS).

The major benefit of the approach proposed in (Cuzzocrea et al., 2006; Cuzzocrea et al., 2007) is a sensible improvement of visualization and exploration activities on high-dimensional spaces via enabling the user to access and browse sub-partitions of these spaces based on semantics rather than on any other arbitrary partitioning scheme, due to the fact that, during interaction, users are typically interested on specific portions of the overall data domain rather than in the entire domain. On the practical plane, Cuzzocrea et al. (2006; 2007) show that while the compression performance of H-IQTS is comparable with state-of-the-art histogram-based data cube compression techniques, the visualization performance of H-IQTS is several orders of magnitude higher than the one of comparison techniques.

FUTURE TRENDS

OLAP Visualization research is still in its preliminary stage, and a lot of work must be done in this field. A key point for the success of this branch of OLAP research is represented by the relevant range of applicability of Visual OLAP in a plethora of real-life, leading applications such as real-time monitoring of multiple streaming data sources and visualization of results produced by advanced *Knowledge Discovery* tools including clustering, association rule discovery, frequent item set mining, sub-graph mining etc.

Future research directions for OLAP Visualization can be identified in the following three main themes: (*i*) *integration with data warehouse management systems*, which will allow us to complete the overall knowledge generation, processing, and visualization experience over multidimensional data sets; (*ii*) *techniques for visualizing integrated data-cube/data-warehouse schemes*, aiming at studying how to visualize multidimensional data domains obtained from the *integration* of multiple and heterogeneous data sources (i.e., how to furnish the BI and DM analyst with an *integrated, unifying visualization metaphor* over heterogeneous cubes?); (*iii*) *visual query languages for multidimensional databases*, aiming at defining a new paradigm able to support intelligent user interaction with multidimensional data, what also poses challenging theoretical foundations on the designing of a powerful *knowledge extraction language*.

CONCLUSION

Similarly to other fundamental issues in OLAP research, such as data cube indexing and compression, the problem of efficiently visualizing OLAP data is an attractive research topic that demands for innovative models and techniques. At present, there are few initiatives encompassing these issues, and intensive work needs to be carried out in this area.

In the spirit of these considerations, in this article we have provided an overview of OLAP Visualization models, issues and techniques, and also critically highlighted advantages and disadvantages of state-of-the-art approaches, while putting in evidence a number of leading applications of these approaches in modern real-life scenarios.

REFERENCES

Ammoura, A., Zaiane, O.R., & Ji, Y. (2001). Towards a Novel OLAP Interface to Distributed Data Warehouses. *Proceedings of the 3rd International Conference on Data Warehousing and Knowledge Discovery*, LNCS Vol. 2114, 174-185.

Chaudhuri, S., & Dayal, U. (1997). An Overview of Data Warehousing and OLAP Technology. *ACM SIGMOD Record*, 26(1), 65-74.

Choong, Y.W., Laurent, D., & Marcel, P. (2003). Computing Appropriate Representations for Multidimensional Data. *Data & Knowledge Engineering*, 45(2), 181-203.

Codd, E.F., Codd, S.B., & Salley, C.T. (1993). Providing OLAP to User-Analysts: An IT Mandate. *E.F. Codd and Associates Technical Report*.

Cuzzocrea, A. Saccà, D., & Serafino, P. (2006). A Hierarchy-Driven Compression Technique for Advanced OLAP Visualization of Multidimensional Data Cubes. *Proceedings of the 8th International Conference on Data Warehousing and Knowledge Discovery*, LNCS Vol. 4081, 106-119.

Cuzzocrea, A., Saccà, D., & Serafino, P. (2007). Semantics-aware Advanced OLAP Visualization of Multidimensional Data Cubes. *International Journal of Data Warehousing and Mining*, 3(4), 1-30.

Eick, S.G. (2000). Visualizing Multi-Dimensional Data. *ACM SIGGRAPH Computer Graphics*, 34(1), 61-67.

Gebhardt, M., Jarke, M., & Jacobs, S. (1997). A Toolkit for Negotiation Support Interfaces to Multi-Dimensional Data. *Proceedings of the 1997 ACM International Conference on Management of Data*, 348-356.

Gray, J., Chaudhuri, S., Bosworth, A., Layman, A., Reichart, D., & Venkatrao, M. (1997). Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals. *Data Mining and Knowledge Discovery*, 1(1), 29-53.

Han, J., & Kamber, M. (2000). *Data Mining: Concepts and Techniques*. Morgan Kauffmann Publishers.

Hanrahan, P., Stolte, C., & Mackinlay, J. (2007). Visual Analysis for Everyone: Understanding Data Exploration and Visualization. *Tableau Software Inc., White Paper*.

Inselberg, A. (2001). Visualization and Knowledge Discovery for High Dimensional Data. *Proceedings of 2nd IEEE UIDIS International Workshop*, 5-24.

Keim, D.A., & Kriegel, H.-P. (1994). VisDB: Database Exploration using Multidimensional Visualization. *IEEE Computer Graphics and Applications*, 14(5), 40-49.

Lee, H.-Y., & Ong, H.-L. (1995). A New Visualisation Technique for Knowledge Discovery in OLAP. *Proceedings of the 1st International Workshop on Integration of Knowledge Discovery in Databases with Deductive and Object-Oriented Databases*, 23-25.

Maniatis, A.S., Vassiliadis, P., Skiadopoulos, S., & Vassiliou, Y. (2003a). Advanced Visualization for OLAP. *Proceedings of the 6th ACM International Workshop on Data Warehousing and OLAP*, 9-16.

Maniatis, A.S., Vassiliadis, P., Skiadopoulos, S., & Vassiliou, Y. (2003b). CPM: A Cube Presentation Model for OLAP. *Proceedings of the 5th International Conference on Data Warehousing and Knowledge Discovery*, LNCS Vol. 2737, 4-13.

Mansmann, S., & Scholl, M.H. (2007). Exploring OLAP Aggregates with Hierarchical Visualization Techniques. *Proceedings of the 22nd Annual ACM Symposium on Applied Computing, Multimedia & Visualization Track*, 1067-1073.

Russom, P. (2000). Trends in Data Visualization Software for Business Users. *DM Review*, May 2000 Issue.

Sifer, M. (2003). A Visual Interface Technique for Exploring OLAP Data with Coordinated Dimension Hierarchies. *Proceedings of the 12th International Conference on Information and Knowledge Management*, 532-535.

Stolte, C., Tang, D., & Hanrahan, P. (2002). Polaris: A System for Query, Analysis, and Visualization of Multidimensional Relational Databases. *IEEE Transactions on Visualization and Computer Graphics*, 8(1), 52-65.

Stolte, C., Tang, D., & Hanrahan, P. (2003). Multiscale Visualization using Data Cubes. *IEEE Transactions on Visualization and Computer Graphics*, 9(2), 176-187.

Tegarden, D.P. (1999). Business Information Visualization. *Communications of the AIS*, 1(1), Article 4.

Techapichetvanich, K., & Datta, A. (2005). Interactive Visualization for OLAP. *Proceedings of the International Conference on Computational Science and its Applications (Part III)*, 206-214.

KEY TERMS

Data Visualization: The use of computer-supported, interactive, visual representations of abstract data to reinforce cognition, hypothesis building and reasoning, building on theory in information design, computer graphics, human-computer interaction and cognitive science.

Multidimensional Cube (Hypercube): A logical structure for fast data analysis based on re-arranging data into a multidimensional array storing numeric facts called measures within array cells, which are indexed by the values drawn from a set of descriptive dimensions.

Multidimensional Visualization: Visualization techniques seeking to efficiently encode more than three dimensions of information simultaneously in a two (three)-dimensional display for multidimensional data analysis purposes.

Online Analytical Processing (OLAP): A methodology for representing, managing and querying massive DW data according to multidimensional and multi-resolution abstractions of them.

Online Transaction Processing (OLTP): A methodology for representing, managing and querying DB data generated by user/application transactions according to flat (e.g., relational) models.

Pivot Table/Cross-Tab: A standard interface for analyzing OLAP data by arranging the specified set of dimensions and measures to obtain the associated totals and subtotals in a two-dimensional (possibly nested) spreadsheet view.

Visual OLAP: An umbrella term encompassing a new generation of end-user tools for interactive ad-hoc exploration of large volumes of multidimensional data via providing a comprehensive framework of advanced visualization techniques for representing retrieved data set, along with a powerful navigation and interaction scheme for specifying, refining, and manipulating subsets of interest.

Online Analytical Processing Systems

Rebecca Boon-Noi Tan
Monash University, Australia

INTRODUCTION

Since its origin in the 1970's research and development into databases systems has evolved from simple file storage and processing systems to complex relational databases systems, which have provided a remarkable contribution to the current trends or environments. Databases are now such an integral part of day-to-day life that often people are unaware of their use. For example, purchasing goods from the local supermarket is likely to involve access to a database. In order to retrieve the price of the item, the application program will access the product database. A database is a collection of related data and the database management system (DBMS) is software that manages and controls access to the database (Elmasri & Navathe, 2004).

BACKGROUND

Data Warehouse

A data warehouse is a specialized type of database. More specifically, a data warehouse is a "repository (or archive) of information gathered from multiple sources, stored under a unified schema, at a single site" (Silberschatz, Korth, & Sudarshan, 2002, p. 843). Chaudhuri and Dayal (1997) consider that a data warehouse should be separately maintained from the organization's operational database since the functional and performance requirements of online analytical processing (OLAP) supported by data warehouses are quite different from those of the online transaction processing (OLTP) traditionally supported by the operational database.

OLAP Versus OLTP

Two reasons why traditional OLTP is not suitable for data warehousing are presented: (a) Given that operational databases are finely tuned to support known OLTP workloads, trying to execute complex OLAP queries against the operational databases would result in unacceptable performance. Furthermore, decision support requires data that might be missing from the operational databases; for instance, understanding trends or making predictions requires historical data, whereas operational databases store only current data. (b) Decision support usually requires consolidating data from many heterogeneous sources: these might include external sources such as stock market feeds, in addition to several operational databases. The different sources might contain data of varying quality, or use inconsistent representations, codes and formats, which have to be reconciled.

Traditional Online Transaction Processing (OLTP)

Traditional relational databases have been used primarily to support OLTP systems. The transactions in an OLTP system usually retrieve and update a small

Figure 1. OLTP system

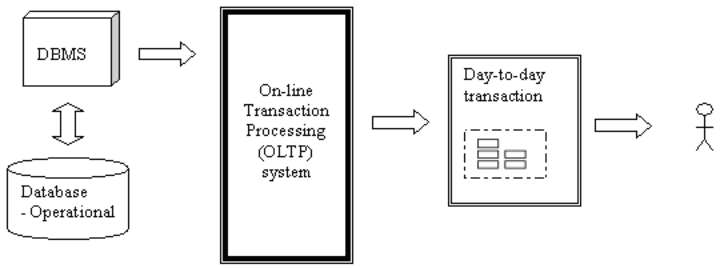
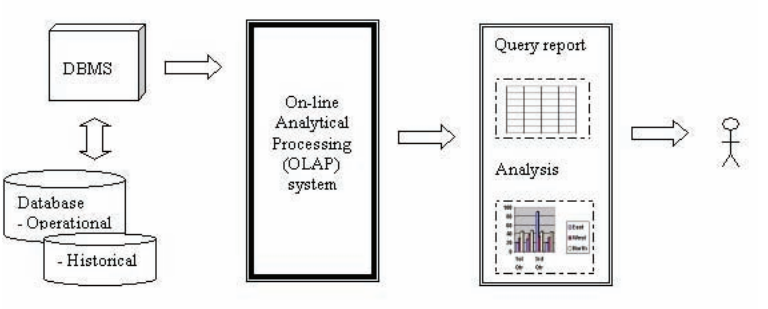


Figure 2. OLAP system



number of records accessed typically on their primary keys. Operational databases tend to be hundreds of megabytes to gigabytes in size and store only current data (Ramakrishnan & Gehrke, 2003).

Figure 1 shows a simple overview of the OLTP system. The operational database is managed by a conventional relational DBMS. OLTP is designed for day-to-day operations. It provides a real-time response. Examples include Internet banking and online shopping.

Online Analytical Processing (OLAP)

OLAP is a term that describes a technology that uses a multi-dimensional view of aggregate data to provide quick access to strategic information for the purposes of advanced analysis (Ramakrishnan & Gehrke, 2003).

OLAP supports queries and data analysis on aggregated databases built in data warehouses. It is a system for collecting, managing, processing and presenting multidimensional data for analysis and management purposes (Figure 2). There are two main implementation methods to support OLAP applications: relational OLAP (ROLAP) and multidimensional OLAP (MOLAP).

ROLAP

Relational online analytical processing (ROLAP) provides OLAP functionality by using relational databases and familiar relational query tools to store and analyse multidimensional data (Ramakrishnan & Gehrke, 2003). Entity Relationship diagrams and normalization techniques are popularly used for database design in OLTP environments. However, the database designs recommended by ER diagrams are inappropriate for decision support systems where efficiency in querying and in loading data (including incremental loads) are crucial. A special schema known as a star schema is used in an OLAP environment for performance reasons (Martyn, 2004). This star schema usually consists of a single fact table and a dimension table for each dimension (Figure 3).

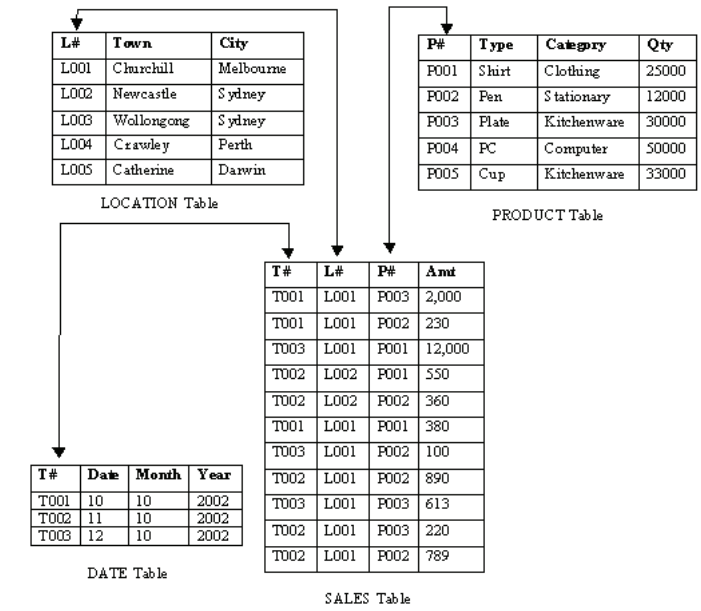
MOLAP

Multidimensional online analytical processing (MOLAP) extends OLAP functionality to multidimensional database management systems (MDBMSs). A MDBMS uses special proprietary techniques to store data in matrix-like n-dimensional arrays (Ramakrishnan & Gehrke, 2003).

The multi-dimensional data cube is implemented by the arrays with the dimensions forming the axes of the cube (Sarawagi, 1997). Therefore, only the data value corresponding to a data cell is stored as direct mapping. MOLAP servers have excellent indexing properties due to the fact that looking for a cell is simple array lookups rather than associative lookups in tables. But unfortunately it provides poor storage utilization, especially when the data set is sparse.

In a multi-dimensional data model, the focal point is on a collection of numeric measures. Each measure depends on a set of dimensions. For instance, the measure attribute is amt as shown in Figure 4. Sales information is being arranged in a three-dimensional array of amt. Figure 4 shows that the array only shows the values for single L# value where L# = L001, which presented as a slice orthogonal to the L# axis.

Figure 3. Star schema showing that location, product, date, and sales are represented as relations



L#	Town	City
L001	Churchill	Melbourne
L002	Newcastle	Sydney
L003	Wollongong	Sydney
L004	Crawley	Perth
L005	Catherine	Darwin

LOCATION Table

P#	Type	Category	Qty
P001	Shirt	Clothing	25000
P002	Pen	Stationary	12000
P003	Plate	Kitchenware	30000
P004	PC	Computer	50000
P005	Cup	Kitchenware	33000

PRODUCT Table

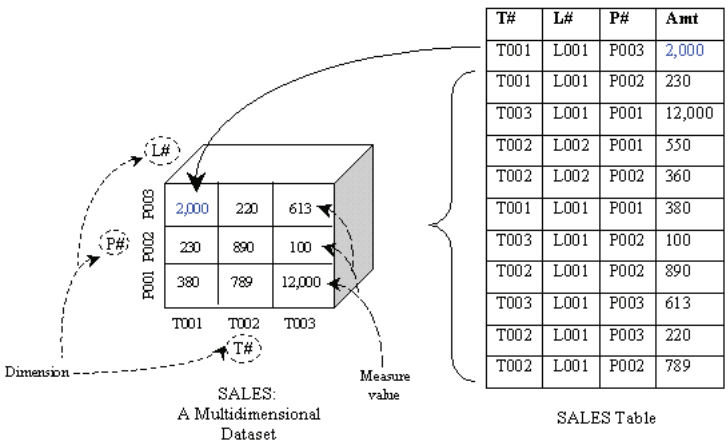
T#	L#	P#	Amt
T001	L001	P003	2,000
T001	L001	P002	230
T003	L001	P001	12,000
T002	L002	P001	550
T002	L002	P002	360
T001	L001	P001	380
T003	L001	P002	100
T002	L001	P002	890
T003	L001	P003	613
T002	L001	P003	220
T002	L001	P002	789

SALES Table

T#	Date	Month	Year
T001	10	10	2002
T002	11	10	2002
T003	12	10	2002

DATE Table

Figure 4. SALES presented as a multidimensional dataset



T#	L#	P#	Amt
T001	L001	P003	2,000
T001	L001	P002	230
T003	L001	P001	12,000
T002	L002	P001	550
T002	L002	P002	360
T001	L001	P001	380
T003	L001	P002	100
T002	L001	P002	890
T003	L001	P003	613
T002	L001	P003	220
T002	L001	P002	789

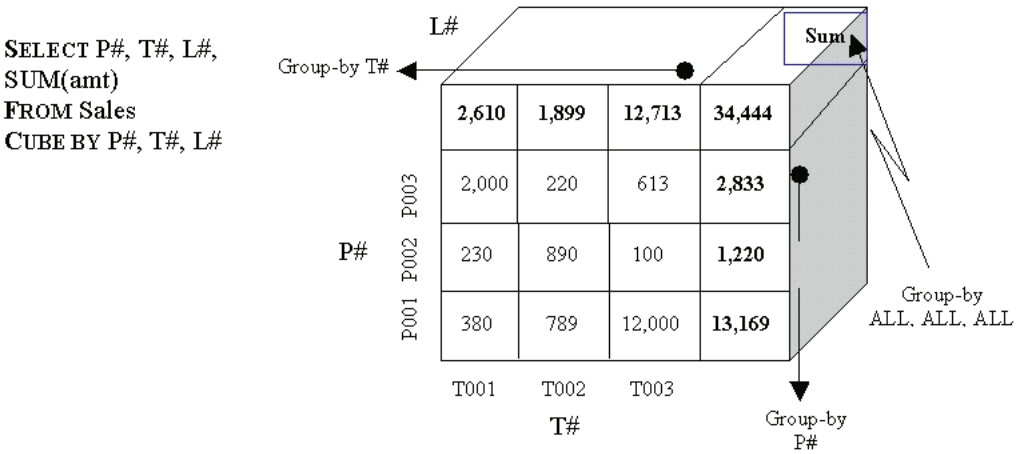
SALES Table

Cube-By Operator

In decision support database systems, aggregation is a commonly used operation. As previously mentioned, current SQL can be very inefficient. Thus, to effectively support decision support queries in OLAP environment, a new operator, Cube-by was proposed by (Gray, Bosworth, Lyaman, & Pirahesh, 1996). It is an extension of the relational operator Group-by. The Cube-by operator computes Group-by corresponding to all possible combinations of attributes in the Cube-by clause.

In order to see how a data cube is formed, an example is provided in Figure 5. It shows an example of data cube formation through executing the cube statement at the top left of the figure. Figure 5 presents two way of presenting the aggregated data: (a) a data cube, and (b) a 3D data cube in table form.

Figure 5. An example of data cube formation through executing the cube statement at the top left of the figure



(a) An example of data cube.

T#	L#	P#	Amt
T001	L001	P003	2,000
T001	L001	P002	230
T003	L001	P001	12,000
T001	L001	P001	380
T003	L001	P002	100
T002	L001	P002	890
T003	L001	P003	613
T002	L001	P003	220
T002	L001	P002	789
T002	L002	P001	550
T002	L002	P002	360

CUBE

T#	L#	P#	Amt
T001	L001	P003	2,000
T001	L001	P002	230
T001	L001	P001	380
T001	L001	ALL	**2,610**
T002	L001	P002	890
T002	L001	P003	220
T002	L001	P002	789
T002	L001	ALL	**1,899**
T003	L001	P001	12,000
T003	L001	P002	100
T003	L001	P003	613
T003	L001	ALL	**12,713**
ALL	L001	P001	**13,169**
ALL	L001	P002	**1,220**
ALL	L001	P003	**2,833**
ALL	L001	ALL	**34,444**
T002	L002	P001	550
T002	L002	P002	360
ALL	L002	P001	550
ALL	L002	P002	360
ALL	L002	ALL	910
T002	L002	ALL	910
ALL	L002	ALL	1,820
ALL	ALL	ALL	36,264

(b) A 3D data cube table (right) built from the table at the left.

Figure 6. Two interesting areas of the processing level in OLAP environment

MAIN THRUST

Processing Level in OLAP Systems

The processing level where the execution of the raw data or aggregated data takes place before presenting the result to the user is shown in Figure 6. The user is only interested in the data in the report in a fast response. However, the background of the processing level is needed in order to provide the user with the result of the aggregated data in an effective and efficient way.

The processing level has been broken down into two distinct areas: *back-end processing* and *front-end processing*. Back-end processing basically deals with the raw data, which is stored in either tables (ROLAP) or arrays (MOLAP) and then process it into aggregated data, which is presented to the user. Front-end processing is computing the raw data and managing the pre-computed aggregated data in either in 3D data cube or n-dimensional table.

It is important to mention that front-end processing is similar to back-end processing as it also deals with raw data. It needs to construct or execute the raw data into the data cube and then store the pre-computed aggregated data in either the memory or disk. It is convenient for the user if the pre-computed aggregated data is ready and is stored in the memory whenever the user requests for it.

The difference between the two processing levels is that the front-end has the pre-computed aggregated data in the memory which is ready for the user to use or analyze it at any given time. On the other hand the back-end processing computes the raw data directly whenever there is a request from the user. This is why the front-end processing is considered to present the aggregated data faster or more efficiently than back-end processing. However, it is important to note that back-end processing and front-end processing are basically one whole processing. The reason behind this break down of the two processing levels is because the problem can be clearly seen in back-end processing and front-end processing. In the next section, the problems associated with each areas and the related work in improving the problems will be considered.

Back-End Processing

Back-end processing involves basically dealing with the raw data, which is stored in either tables (ROLAP) or arrays (MOLAP) as shown in Figure 7. The user queries the raw data for decision-making purpose. The raw data is then processed and computed into aggregated data, which will be presented to the user for analyzing. Generally the basic stage: *extracting,* is followed by two sub-stages: *indexing,* and *partitioning* in back-end processing.

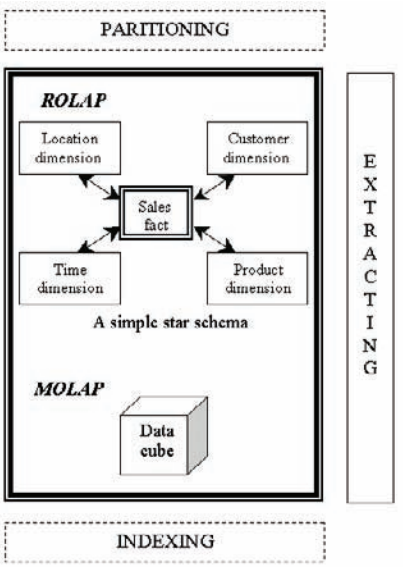
Extracting is the process of querying the raw data either from tables (ROLAP) or arrays (MOLAP) and computing it. The process of extracting is usually time-consuming. Firstly, the database (data warehouse) size is extremely large and secondly the computation time is equally high. However, the user is only interested in a fast response time for providing the resulted data. Another consideration is that the analyst or manager using the data warehouse may have time constraints. There are two sub-stages or fundamental methods of handling the data: (a) indexing and (b) partitioning in order to provide the user with the resulted data in a reasonable time frame.

Indexing has existed in databases for many decades. Its access structures have provided faster access to the base data. For retrieval efficiency, index structures would typically be defined especially in data warehouses or ROLAP where the fact table is very large (Datta, VanderMeer & Ramamritham, 2002). O'Neil and Quass (1997) have suggested a number of important indexing schemes for data warehousing including bitmap index, value-list index, projection index, data index. Data index is similar to projection index but it exploits a positional indexing strategy (Datta, VanderMeer, & Ramamritham, 2002). Interestingly MOLAP servers have better indexing properties than ROLAP servers since they look for a cell using simple array lookups rather than associative lookups in tables.

Partitioning of raw data is more complex and challenging in data warehousing as compared to that of relational and object databases. This is due to the several choices of partitioning of a star schema (Datta, VanderMeer, & Ramamritham, 2002). The data fragmentation concept in the context of distributed databases aims to reduce query execution time and facilitate the parallel execution of queries (Bellatreche, Karlapalem, Mohania, & Schneide, 2000).

In a data warehouse or ROLAP, either the dimension tables or the fact table or even both can be fragmented. Bellatreche et al. (2000) have proposed a methodology for applying the fragmentation techniques in a data warehouse star schema to reduce the total query

Figure 7. Stages in back-end processing



execution cost. The data fragmentation concept in the context of distributed databases aims to reduce query execution time and facilitates the parallel execution of queries.

Front-End Processing

Front-end processing is computing the raw data and managing the pre-computed aggregated data in either in 3D data cube or n-dimensional table. The three basic types of stages that are involved in Front-end processing are shown in Figure 8. They are (i) constructing, (ii) storing and (iii) querying. Constructing is basically the computation process of a data cube by using cube-by operator. Cube-by is an expensive approach, especially when the number of Cube-by attributes and the database size are large.

The difference between constructing and extracting needs to be clearly defined. Constructing in the Front-end processing is similar to the extracting in the Back-end processing as both of them are basically querying the raw data. However, in this case, extracting concentrates on how the fundamental methods can help in handling the raw data in order to provide efficient retrieval. Constructing in the Front-end processing concentrates on the cube-by operator that involves the computation of raw data.

Storing is the process of putting the aggregated data into either the n-dimension table of rows or the n-dimensional arrays or data cube. There are two parts within the storing process. One part is to store temporary raw data in the memory for executing purpose and other part is to store pre-computed aggregated data. Hence, there are two problems related to the storage: (a) insufficient memory space – due to the loaded raw data and also in addition to the incremental loads; (b) poor storage utilization – the array may not fit into memory especially when the data set is sparse.

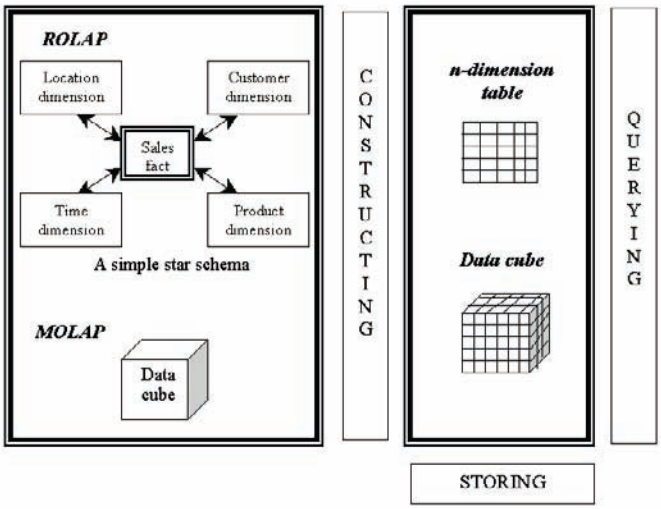
Querying is the process of extracting useful information from the pre-computed data cube or n-dimensional table for decision makers. It is important to take note that Querying is also part of Constructing process. Querying also makes use of cube-by operator that involves the computation of raw data. ROLAP is able to support ad hoc requests and allows unlimited access to dimensions unlike MOLAP only allows limited access to predefined dimensions. Despite the fact that it is able to support ad hoc requests and access to dimensions, certain queries might be difficult to fulfill the need of decision makers and also to reduce the querying execution time when there is n-dimensional query as the time factor is important to decision makers.

To conclude, the three stages and their associated problems in the OLAP environment have been outlined. First, Cube-by is an expensive approach, especially when the number of Cube-by attributes and the database size are large. Second, storage has insufficient memory space – this is due to the loaded data and also in addition to the incremental loads. Third, storage is not properly utilized – the array may not fit into memory especially when the data set is sparse. Fourth, certain queries might be difficult to fulfill the need of decision makers. Fifth, the execution querying time has to be reduced when there is n-dimensional query as the time factor is important to decision makers. However, it is important to consider that there is scope for other possible problems to be identified.

FUTURE TRENDS

Problems have been identified in each of the three stages, which have generated considerable attention from researchers to find solutions. Several researchers have proposed a number of algorithms to solve these cube-by problems. Examples include:

Figure 8. Stages in front-end processing



Constructing

Cube-by operator is an expensive approach, especially when the number of Cube-by attributes and the database size are large.

Fast Computation Algorithms

There are algorithms aimed at fast computation of large sparse data cube (Ross & Srivastava, 1997; Beyer & Ramakrishnon, 1999). Ross and Srivastava (1997) have taken into consideration the fact that real data is frequently sparse. (Ross & Srivastava, 1997) partitioned large relations into small fragments so that there was always enough memory to fit in the fragments of large relation. Whereas, Beyer and Ramakrishnan (1999) proposed the bottom-up method to help reduce the penalty associated with the sorting of many large views.

Parallel Processing System

The assumption of most of the fast computation algorithms is that their algorithms can be applied into the parallel processing system. Dehne, Eavis, Hambrusch, and Rau-Chaplin. (2002) presented a general methodology for the efficient parallelization of exiting data cube construction algorithms. Their paper described two different partitioning strategies, one for top-down and one for bottom-up cube algorithms. They provide a good summary and comparison with other parallel data cube computations (Ng, Wagner, & Yin., 2001; Yu & Lu, 2001). In Ng et al. (2001), the approach considers the parallelization of sort-based data cube construction, whereas Yu and Lu (2001) focuses on the overlap between multiple data cube computations in a sequential setting.

Storing

Two problems related to the storage: (a) insufficient memory space; (b) poor storage utilization.

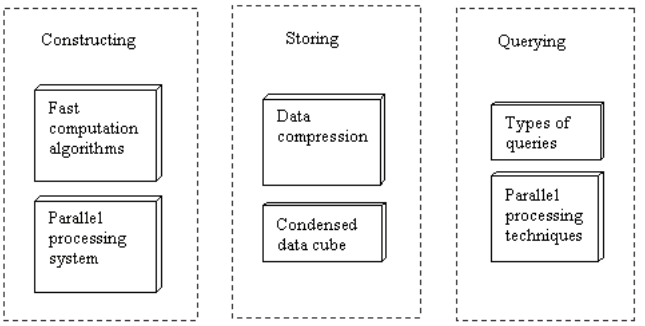
Data Compression

Another technique is related to data compression. Lakshmanan, Pei, and Zhao (2003) have developed a systematic approach to achieve efficacious data cube construction and exploration by semantic summarization and compression.

Condensed Data Cube

Wang, Feng, Lu, and Yu (2002) have proposed a new concept called a condensed data cube. This new approach reduces the size of data cube and hence its computation time. They make use of “single base tuple” compression to generate a “condensed cube” so it is smaller in size.

Figure 9. Working areas in the OLAP system



Querying

It might be difficult for certain queries to fulfill both the need of the decision makers and also the need to reduce the querying execution time when there is n-dimensional query as the time factor is important to decision makers.

Types of Queries

Other work has focused on specialized data structures for fast processing of special types of queries. Lee, Ling and Li (2000) have focused on range-max queries and have proposed hierarchical compact cube to support the range-max queries. The hierarchical structure stores not only the maximum value of all the children sub-cubes, but also stores one of the locations of the maximum values among the children sub-cubes. On the other hand, Tan, Taniar, and Lu (2004) focus on data cube queries in term of SQL and have presented taxonomy of data cube queries. They also provide a comparison with different type queries.

Parallel Processing Techniques

Taniar and Tan (2002) have proposed three parallel techniques to improve the performance of the data cube queries. However, the challenges in this area continue to grow with the cube-by problems.

CONCLUSION

In this overview, the OLAP systems in general have been considered, followed by processing level in the OLAP systems and lastly the related and future work in the OLAP systems. A number of problems and solutions in the OLAP environment have been presented. However, consideration needs to be given to the possibility that other problems maybe identified which in turn will present new challenges for the researchers to address.

REFERENCES

Bellatreche, L., Karlapalem, K., Mohania M., & Schneider M. (2000, September). What can partitioning do for your data warehouses and data marts? *International IDEAS conference* (pp. 437-445), Yokohoma, Japan.

Beyer, K.S., & Ramakrishnan, R. (1999, June). Bottom-up computation of sparse and iceberg cubes. *International ACM SIGKDD conference* (pp. 359-370), Philadelphia, PA.

Chaudhuri, S., & Dayal, U. (1997). An overview of data warehousing and OLAP technology. *ACM SIGMOD Record, 26*, 65-74.

Datta, A., VanderMeer, D., & Ramamritham, K. (2002). Parallel Star Join + DataIndexes: Efficient Query Processing in Data Warehouses and OLAP. *IEEE*

Transactions on Knowledge & Data Engineering, 14(6), 1299-1316.

Dehne, F., Eavis, T., Hambrusch, S., & Rau-Chaplin, A. (2002). Parallelizing the data cube. *International Journal of Distributed & Parallel Databases, 11*, 181-201.

Elmasri, R., & Navathe, S.B. (2004). *Fundamentals of database systems*. Boston, MA: Addison Wesley.

Gray, J., Bosworth, A., Lyaman, A. & Pirahesh, H. (1996, June). Data cube: A relational aggregation operator generalizing GROUP-BY, CROSS-TAB, and SUB-TOTALS. *International ICDE Conference* (pp. 152-159), New Orleans, Louisiana.

Lakshmanan, L.V.S., Pei, J., & Zhao, Y. (2003, September). Efficacious data cube exploration by semantic semmarization and compression. *International VLDB Conference* (pp. 1125-1128), Berlin, Germany.

Lee, S.Y, Ling, T.W., & Li, H.G. (2000, September). Hierarchical compact cube for range-max queries. *International VLDB Conference* (pp. 232-241), Cairo, Egypt.

Martyn, T. (2004). Reconsidering multi-dimensional schemas. *ACM SIGMOD Record*, 83-88.

Ng, R.T., Wagner, A., & Yin, Y. (2001, May). Iceberg-cube computation with pc clusters. *International ACM SIGMOD Conference* (pp. 25-36), Santa Barbara, California.

O'Neil, P., & Graefe, G. (1995). Multi-table joins through bit-mapped join indices. *SIGMOD Record, 24*(3), 8-11.

Ramakrishnan, R., & Gehrke, J. (2003). *Database management systems*. NY: McGraw-Hill.

Ross, K.A., & Srivastava, D. (1997, August). Fast computation of sparse datacubes. *International VLDB Conference* (pp. 116-185), Athens, Greece.

Sarawagi, S. (1997). Indexing OLAP data. *IEEE Data Engineering Bulletin, 20*(1), 36-43.

Silberschatz, A., Korth, H., & Sudarshan, S. (2002). *Database system concepts*. NY: McGraw-Hill.

Taniar, D., & Tan, R.B.N. (2002, May). Parallel processing of multi-join expansion-aggregate data cube query in high performance database systems. *International I-SPAN Conference* (pp. 51-58), Manila, Philippines.

Tan, R.B.N., Taniar, D., & Lu, G.J. (2004). A Taxonomy for Data Cube Queries, *International Journal for Computers and Their Applications, 11*(3), 171-185.

Wang, W., Feng, J.L., Lu, H.J., & Yu, J.X. (2002, February). Condensed Cube: An effective approach to reducing data cube size. *International Data Engineering Conference* (pp. 155-165), San Jose, California.

Yu, J.X., & Lu, H.J. (2001, April). Multi-cube computation. *International DASFAA Conference* (pp. 126-133), Hong Kong, China.

KEY TERMS

Back-End Processing: Is dealing with the raw data, which is stored in either tables (ROLAP) or arrays (MOLAP).

Data Cube Operator: Computes Group-by corresponding to all possible combinations of attributes in the Cube-by clause.

Data Warehouse: A type of database A "subject-oriented, integrated, time-varying, non-volatile collection of data that issued primarily in organizational decision making" (Elmasri & Navathe, 2004, p.900).

Front-End Processing: Is computing the raw data and managing the pre-computed aggregated data in either in 3D data cube or n-dimensional table.

Multidimensional OLAP (MOLAP): Extends OLAP functionality to multidimensional database management systems (MDBMSs).

Online Analytical Processing (OLAP): Is a term used to describe the analysis of complex data from data warehouse (Elmasri & Navathe, 2004, p.900).

Relational OLAP (ROLAP): Provides OLAP functionality by using relational databases and familiar relational query tools to store and analyse multidimensional data.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 523-528, copyright 2005 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

Online Signature Recognition

Indrani Chakravarty
Indian Institute of Technology, India

Nilesh Mishra
Indian Institute of Technology, India

Mayank Vatsa
Indian Institute of Technology, India

Richa Singh
Indian Institute of Technology, India

P. Gupta
Indian Institute of Technology, India

INTRODUCTION

Security is one of the major issues in today's world and most of us have to deal with some sort of passwords in our daily lives; but, these passwords have some problems of their own. If one picks an easy-to-remember password, then it is most likely that somebody else may guess it. On the other hand, if one chooses too difficult a password, then he or she may have to write it somewhere (to avoid inconveniences due to forgotten passwords) which may again lead to security breaches. To prevent passwords being hacked, users are usually advised to keep changing their passwords frequently and are also asked not to keep them too trivial at the same time. All these inconveniences led to the birth of the biometric field. The verification of handwritten signature, which is a behavioral biometric, can be classified into off-line and online signature verification methods. Online signature verification, in general, gives a higher verification rate than off-line verification methods, because of its use of both static and dynamic features of the problem space in contrast to off-line which uses only the static features. Despite greater accuracy, online signature recognition is not that prevalent in comparison to other biometrics. The primary reasons are:

- It cannot be used everywhere, especially where signatures have to be written in ink; e.g. on cheques, only off-line methods will work.
- Unlike off-line verification methods, online methods require some extra and special hardware, e.g. electronic tablets, pressure sensitive signature pads, etc. For off-line verification method, on the other hand, we can do the data acquisition with optical scanners.
- The hardware for online are expensive and have a fixed and short life cycle.

In spite of all these inconveniences, the use online methods is on the rise and in the near future, unless a process requires particularly an off-line method to be used, the former will tend to be more and more popular.

BACKGROUND

Online verification methods can have an accuracy rate of as high as 99%. The reason behind is its use of both static and dynamic (or temporal) features, in comparison to the off-line, which uses only the static features (Ramesh & Murty, 1999). The major differences between off-line and online verification methods do not lie with only the feature extraction phases and accuracy rates, but also in the modes of data acquisition, preprocessing and verification/recognition phases, though the basic sequence of tasks in an online verification (or recognition) procedure is exactly the same as that of the off-line. The phases that are involved comprise of:

- Data Acquisition
- Preprocessing and Noise Removal
- Feature Extraction and
- Verification (or Identification)

However, online signatures are much more difficult to forge than off-line signatures (reflected in terms of higher accuracy rate in case of online verification methods), since online methods involve the dynamics of the signature such as the pressure applied while writing, pen tilt, the velocity with which the signature is done etc. In case of off-line, the forger has to copy only the shape (Jain & Griess, 2000) of the signature. On the other hand, in case of online, the hardware used captures the dynamic features of the signature as well. It is extremely difficult to deceive the device in case of dynamic features, since the forger has to not only copy the characteristics of the person whose signature is to be forged, but also at the same time, he has to hide his own inherent style of writing the signature. There are four types of forgeries: random, simple, skilled and traced forgeries (Ammar, Fukumura, & Yoshida, 1988; Drouhard, Sabourin, & Godbout, 1996). In case of online signatures, the system shows almost 100% accuracy for the first two classes of forgeries and 99% in case of the latter. But, again, a forger can also use a compromised signature-capturing device to repeat a previously recorded signature signal. In such extreme cases, even online verification methods may suffer from repetition attacks when the signature-capturing device is not physically secure.

MAIN THRUST

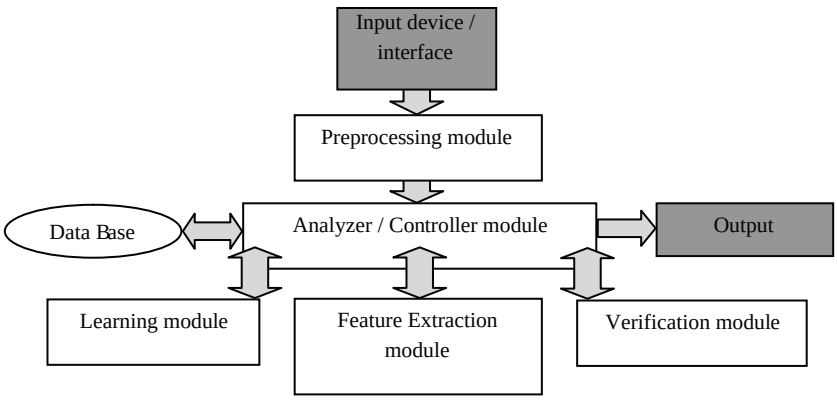
Although the basic sequence of tasks in online signature verification is almost the same as that of off-line methods, the modes differ from each other especially in the ways the data acquisition, preprocessing and feature extraction are carried out. More specifically, the sub-modules of online are much more difficult with respect to off-line (Jain & Griess, 2000). Figure 1 gives a generic structure of an online signature verification system. The online verification system can be classified into the following modules:

- Data Acquisition,
- Preprocessing,
- Feature Extraction,
- Learning and Verification.

Data Acquisition

Data acquisition (of the dynamic features) in online verification methods is generally carried out using special devices called transducers or digitizers (Tappert, Suen, & Wakahara, 1990, Wessels & Omlin, 2000), in contrast to the use of high resolution scanners in case of off-line. The commonly used instruments include the electronic tablets (which consist of a grid to capture the x and y coordinates of the pen tip movements), pressure sensitive tablets, digitizers involving technologies such as acoustic sensing in air medium, surface acoustic waves, triangularization of reflected laser beams, and optical sensing of a light pen to extract information about the number of strokes, velocity of

Figure 1. Modular structure of a generic online verification system



signing, direction of writing, pen tilt, pressure with which the signature is written etc.

Preprocessing

Preprocessing in online is much more difficult than in off-line, because it involves both noise removal (which can be done using hardware or software) (Plamondon & Lorette, 1989) and segmentation in most of the cases. The other preprocessing steps that can be performed are signal amplifying, filtering, conditioning, digitizing, resampling, signal truncation, normalization, etc. However, the most commonly used include:

- **External Segmentation:** Tappert, Suen and Wakahara (1990) define external segmentation as the process by which the characters or words of a signature are isolated before the recognition is carried out.
- **Resampling:** This process is basically done to ensure uniform smoothing to get rid of the redundant information, as well as to preserve the required information for verification by comparing the spatial data of two signatures. According to Jain and Griess (2000), here, the distance between two critical points is measured, and if the total distance exceeds a threshold called the resampling length(which is calculated by dividing the distance by the number of sample points for that segment), then a new point is created by using the gradient between the two points,
- **Noise Reduction:** Noise is nothing but irrelevant data, usually in the form of extra dots or pixels in images (in case of off-line verification methods) (Ismail & Gad, 2000), which do not belong to the signature, but are included in the image (in case of off-line) or in the signal (in case of online), because of possible hardware problems (Tappert, Suen, & Wakahara, 1990) or presence of background noises like dirt or by faulty hand movements (Tappert, Suen, & Wakahara, 1990) while signing.

  The general techniques used are:

  - **Smoothening:** in which a point is averaged with its neighbors. (Tappert, Sue, & Wakahara, 1990)
  - **Filtering:** Online thinning (Tappert, Suen, & Wakahara, 1990) is not the same as off-line thinning (Baltzakis & Papamarkos, 2001), although, both decrease the number of irrelevant dots or points. According to Tappert, Suen and Wakahara (1990), it can be carried out either by forcing a minimum distance between adjacent points or by forcing a minimum change in the direction of the tangent to the drawing for consecutive points.
  - **Wild Point Correction:** This removes noises (points) caused by hardware problems (Tappert, Suen, & Wakahara, 1990).
  - **Dehooking:** Dehooking on the other hand removes hook like features from the signature, which usually occur at the start or end of the signature (Tappert, Suen, & Wakahara, 1990).
  - **Dot Reduction:** The dot size is reduced to single points. (Tappert, Suen, & Wakahara, 1990)

- **Normalization:** Off-line normalization often limits itself to scaling the image to a standard size and/or removing blank columns and rows. In case of online, normalization includes deskewing, baseline drift correction (orienting the signature to horizontal), size normalization and stroke length normalization (Ramesh & Murty, 1999; Tappert, Suen, & Wakahara, 1990; Wessels & Omlin, 2000).

Feature Extraction

Online signature extracts both the static and the dynamic features. Some of the static and dynamic features have been listed below.

- **Static Features:** Although both static and dynamic information are available to the online verification system, in most of the cases, the static information is discarded, as dynamic features are quite rich and they alone give high accuracy rates. Some of the static features used by online method are:

  - Width and height of the signature parts (Ismail & Gad, 2000)
  - Width to height ratio (Ismail & Gad, 2000)

- Number of cross and edge points (Baltzakis & Papamarkos, 2001)
- Run lengths of each scan of the components of the signature (Xiao & Leedham, 2002)
- Kurtosis (horizontal and vertical), skewness, relative kurtosis, relative skewness, relative horizontal and vertical projection measures (Bajaj & Chaudhury, 1997; Ramesh & Murty, 1999)
- **Envelopes:** upper and lower envelope features (Bajaj & Chaudhury, 1997; Ramesh & Murty, 1999)
- **Alphabet Specific Features:** Like presence of ascenders, descenders, cusps, closures, dots etc. (Tappert, Suen, & Wakahara, 1990)

- **Dynamic Features:** Though online methods utilize some of the static features, they give more emphasis to the dynamic features, since these features are more difficult to imitate.

The most widely used dynamic features include:

- **Position ($u_x(t)$ and $u_y(t)$):** e.g. the pen tip position when the tip is in the air, and when it is in the vicinity of the writing surface etc (Jain & Griess, 2000; Plamondon & Lorette, 1989).
- Pressure ($u_p(t)$) (Jain & Griess, 2000; Plamondon & Lorette, 1989)
- **Forces ($u_f(t)$, $u_{fx}(t)$, $u_{fy}(t)$):** Forces are computed from the position and pressure coordinates (Plamondon & Lorette, 1989).
- **Velocity ($u_v(t)$, $u_{vx}(t)$, $u_{vy}(t)$):** It can be derived from position coordinates (Jain & Griess, 2000; Plamondon & Lorette, 1989).
- Absolute and relative speed between two critical points (Jain, Griess, & Connell, 2002)
- **Acceleration ($u_a(t)$, $u_{ax}(t)$, $u_{ay}(t)$):** Acceleration can be derived from velocity or position coordinates. It can also be computed using an accelerometric pen (Jain & Griess, 2000; Plamondon & Lorette, 1989).
- **Parameters:** A number of parameters like number of peaks, starting direction of the signature, number of pen lifts, means and standard deviations, number of maxima and minima for each segment, proportions, signature path length, path tangent angles etc. are also calculated apart from the above mentioned functions to increase the dimensionality (Gupta, 1997; Plamondon & Lorette, 1989; Wessels & Omlin, 2000). Moreover, all these features can be of both global and local in nature.

Verification and Learning

For online, examples of comparison methods include use of:

- Corner Point and Point to Point Matching algorithms (Zhang, Pratikakis, Cornelis, & Nyssen, 2000)
- Similarity measurement on logarithmic spectrum (Lee, Wu, & Jou, 1998)
- Extreme Points Warping (EPW) (Feng & Wah, 2003)
- String Matching and Common threshold (Jain, Griess, & Connell, 2002)
- Split and Merging, (Lee, Wu, & Jou, 1997)
- Histogram classifier with global and local likeliness coefficients (Plamondon & Lorette, 1989)
- Clustering analysis (Lorette, 1984)
- Dynamic programming based methods; matching with Mahalanobis pseudo-distance (Sato & Kogure,1982)
- Hidden Markov Model based methods (McCabe, 2000; Kosmala & Rigoll, 1998)

Table 1 gives a summary of some prominent works in the online signature verification field.

Performance Evaluation

Performance evaluation of the output (which is to accept or reject the signature) is done using false rejection rate (FRR or type I error) and false acceptance rate (FAR or type II error) (Huang & Yan, 1997). The values for error rates of different approaches have been included in the comparison table above. Equal error rate (ERR) which is calculated using FRR and FAR is also used for measuring the accuracy of the systems.

GENERAL PROBLEMS

The feature set in online has to be taken very carefully, since it must have sufficient interpersonal variability

Table 1. Summery of some prominent online papers

Author	Mode Of Verification	Database	Feature Extraction	Results (Error rates)
Wu, Lee and Jou (1997)	Split and merging	200 genuine and 246 forged	coordinates to represent the signature and the velocity	86.5 % accuracy rate for genuine and 97.2 % for forged
Wu, Lee and Jou (1998)	Similarity measurement on logarithmic spectrum	27 people each 10 signs, 560 genuine and 650 forged for testing	Features based on coefficients of the logarithmic spectrum	FRR=1.4 % and FAR=2.8 %
Zhang, Pratikakis, Cornelis and Nyssen (2000)	Corner point matching algorithm and Point to point matching algorithm	188 signatures, from 19 people	corner points extracted based on velocity information	0.1 % mismatch in segments in case of corner point algorithm and 0.4 % mismatch in case of point to point matching algorithm
Jain, Griess and Connell (2002)	String matching and common threshold	1232 signatures of 102 individual	Number of strokes, co-ordinate distance between two points, angle with respect to x and y axis, curvature, distance from centre of gravity, grey value in 9X9 neighborhood and velocity features	Type I: 2.8% Type II: 1.6%
Feng and Wah (2003)	Extreme Points Warping (both Euclidean distance and Correlation Coefficients used)	25 users contributed 30 genuine signatures and 10 forged signatures	x and y trajectories used, apart from torque and center of mass.	EER (Euclidean) = 25.4 % EER (correlation) = 27.7 %

so that the input signature can be classified as genuine or forgery. In addition, it must also have a low intra personal variability so that an authentic signature is accepted. Therefore, one has to be extremely cautious, while choosing the feature set, as increase in dimensionality does not necessarily mean an increase in efficiency of a system.

FUTURE TRENDS

Biometrics is gradually replacing the conventional password and ID based devices, since it is both convenient and safer than the earlier methods. Today, it is not difficult at all to come across a fingerprint scanner or an online signature pad. Nevertheless, it still requires a lot of research to be done to make the system infallible, because even an accuracy rate of 99% can cause failure of the system when scaled to the size of a million.

So, it will not be strange if we have ATMs in near future granting access only after face recognition, fingerprint scan and verification of signature via embedded devices, since a multimodal system will have a lesser chance of failure than a system using a single biometric or a password/ID based device. Currently online and off-line signature verification systems are two disjoint approaches. Efforts must be also made to integrate the two approaches enabling us to exploit the higher accuracy of online verification method and greater applicability of off-line method.

CONCLUSION

Most of the online methods claim near about 99% accuracy for signature verification. These systems are gaining popularity day by day and a number of products are currently available in the market. However, online verification is facing stiff competition from fingerprint verification system which is both more portable and has higher accuracy. In addition, the problem of maintaining a balance between FAR and FRR has to be maintained. Theoretically, both FAR and FRR are inversely related to each other. That is, if we keep tighter thresholds to decrease FAR, inevitably we increase the FRR by rejecting some genuine signatures. Further research is thus still required to overcome this barrier.

REFERENCES

Ammar, M., Fukumura, T., & Yoshida Y. (1988). Off-line preprocessing and verification of signatures. *International Journal of Pattern Recognition and Artificial Intelligence, 2*(4), 589-602.

Ammar, M., Fukumura, T., & Yoshida Y. (1990). Structural description and classification of signature images. *Pattern Recognition, 23*(7), 697-710.

Bajaj, R., & Chaudhury, S. (1997). Signature verification using multiple neural classifiers. *Pattern Recognition, 30*(1), l-7.

Baltzakis, H., & Papamarkos, N. (2001). A new signature verification technique based on a two-stage neural network classifier. *Engineering Applications of Artificial Intelligence, 14*, 95-103.

Drouhard, J. P., Sabourin, R., & Godbout, M. (1996). A neural network approach to off-line signature verification using directional pdf. *Pattern Recognition, 29*(3), 415-424.

Feng, H., & Wah, C. C. (2003). Online signature verification using a new extreme points warping technique. *Pattern Recognition Letters 24*(16), 2943-2951.

Gupta, J., & McCabe, A. (1997). *A review of dynamic handwritten signature verification.* Technical Article, James Cook University, Australia.

Huang, K., & Yan, H. (1997). Off-line signature verification based on geometric feature extraction and neural network classification. *Pattern Recognition, 30*(1), 9-17.

Ismail, M.A., & Gad, S. (2000). Off-line Arabic signature recognition and verification. *Pattern Recognition, 33*, 1727-1740.

Jain, A. K., & Griess, F. D. (2000). *Online signature verification.* Project Report, Department of Computer Science and Engineering, Michigan State University, USA.

Jain, A. K., Griess, F. D., & Connell, S. D. (2002). Online signature verification. *Pattern Recognition, 35*(12), 2963-2972.

Kosmala, A., & Rigoll, G. (1998). A systematic comparison between online and off-line methods for signature verification using hidden markov models. *14th International Conference on Pattern Recognition* (pp. 1755-1757).

Lee, S. Y., Wu, Q. Z., & Jou, I. C. (1997). Online signature verification based on split and merge matching mechanism. *Pattern Recognition Letters, 18*, 665-673

Lee, S. Y., Wu, Q. Z., & Jou, I. C. (1998). Online signature verification based on logarithmic spectrum. *Pattern Recognition, 31*(12), 1865-1871

Lorette, G. (1984). Online handwritten signature recognition based on data analysis and clustering. *Proceedings of 7th International Conference on Pattern Recognition, Vol. 2* (pp. 1284-1287).

McCabe, A. (2000). Hidden markov modeling with simple directional features for effective and efficient handwriting verification. *Proceedings of the Sixth Pacific Rim International Conference on Artificial Intelligence*.

Plamondon R., & Lorette, G. (1989). Automatic signature verification and writer identification – the state of the art. *Pattern Recognition, 22*(2), 107-131.

Ramesh, V.E., & Murty, M. N. (1999). Off-line signature verification using genetically optimized weighted features. *Pattern Recognition, 32*(7), 217-233.

Sato, Y., & Kogure, K. (1982). Online signature verification based on shape, motion and handwriting pressure. *Proceedings of 6th International Conference on Pattern Recognition, Vol. 2* (pp. 823-826).

Tappert, C. C., Suen, C. Y., & Wakahara, T. (1990). The state of the art in on-line handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 12*(8).

Wessels, T., & Omlin, C. W. (2000). A hybrid approach for signature verification. *International Joint Conference on Neural Networks*.

Xiao, X., & Leedham, G. (2002). Signature verification using a modified bayesian network. *Pattern Recognition, 35*, 983-995.

Zhang, K., Pratikakis, I., Cornelis, J., & Nyssen, E. (2000). Using landmarks in establishing a point to point correspondence between signatures. *Pattern Analysis and Applications, 3*, 69-75.

KEY TERMS

Equal Error Rate: The error rate when the proportions of FAR and FRR are equal. The accuracy of the biometric system is inversely proportional to the value of EER.

False Acceptance Rate: Rate of acceptance of a forged signature as a genuine signature by a handwritten signature verification system.

False Rejection Rate: Rate of rejection of a genuine signature as a forged signature by a handwritten signature verification system.

Global Features: The features are extracted using the complete signature image or signal as a single entity.

Local Features: The geometric information of the signature is extracted in terms of features after dividing the signature image or signal into grids and sections.

Online Signature Recognition: The signature is captured through a digitizer or an instrumented pen and both geometric and temporal information are recorded and later used in the recognition process.

Random Forgery: Random forgery is one in which the forged signature has a totally different semantic meaning and overall shape in comparison to the genuine signature.

Simple Forgery: Simple forgery is one in which the semantics of the signature are the same as that of the genuine signature, but the overall shape differs to a great extent, since the forger has no idea about how the signature is done.

Skilled Forgery: In skilled forgery, the forger has a prior knowledge about how the signature is written and practices it well, before the final attempt of duplicating it.

Traced Forgery: For traced forgery, a signature instance or its photocopy is used as a reference and tried to be forged.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 885-890, copyright 2005 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

Ontologies and Medical Terminologies

James Geller
New Jersey Institute of Technology, USA

INTRODUCTION

Ontologies

The term "Ontology" was popularized in Computer Science by Thomas Gruber at the Stanford Knowledge Systems Lab (KSL). Gruber's highly influential papers defined an ontology as "an explicit specification of a conceptualization." (Gruber, 1992; Gruber 1993). Gruber cited a conceptualization as being "the objects and concepts, and other entities that are assumed to exist in some area of interest and the relationships that hold among them." (Genesereth & Nilsson, 1987). The term "Ontology" has been used in computer science at least since (Neches, 1991), but is derived from philosophy where it defines a "systematic account of existence," usually contrasted with "Epistemology."

Gruber's work is firmly grounded in Knowledge Representation and Artificial Intelligence research going back to McCarthy and Hayes classical paper (McCarthy & Hayes, 1969). Gruber's work also builds on frame systems (Minsky, 1975; Fikes and Kehler, 1985) which have their roots in Semantic Networks, pioneered by (Quillian, 1968) and popularized through the successful and widespread KL-ONE family (Brachman & Schmolze, 1985). One can argue that Gruber's ontologies are structurally very close to previous work in frame-based knowledge representation systems. However, Gruber focused on the notion of knowledge sharing which was a popular topic at KSL around the same time, especially in the form of the Knowledge Interchange Format (KIF) (Genesereth, 1991).

Ontologies have recently moved center stage in Computer Science as they are a major ingredient of the Semantic Web (Berners-Lee et al., 2001), the next generation of the World-Wide Web. Ontologies have also been used in Data Mining (see below) and in (database) schema integration.

Medical Terminologies

In the field of Medical Informatics a rich set of Medical Terminologies has been developed over the past twenty years. Many of these terminologies have as their backbone a taxonomy of concepts and IS-A (subclass) relationships. This IS-A hierarchy was pioneered in the semantic networks and frame systems mentioned above. With this structural commonality of ontologies and Medical Terminologies in mind, we will treat both kinds of knowledge representation systems together. Some of the largest existing ontologies have been developed in Medical Informatics, which makes this field especially interesting. For example, the Unified Medical Language System (UMLS; Humphreys et al., 1998) Metathesaurus contains information about over 1.5 million biomedical concepts and 7.2 million concept names.[1]

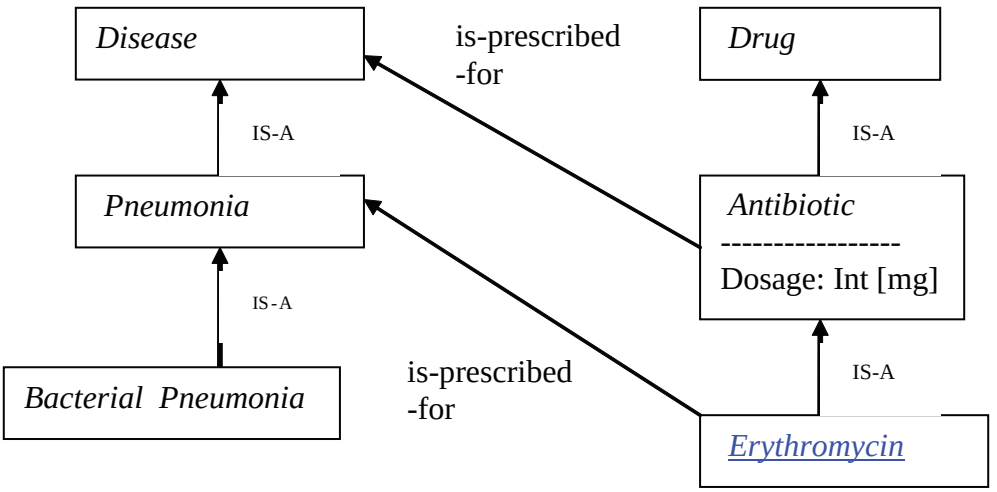
BACKGROUND

The easiest way to understand ontologies is to look at them from a structural perspective. Quillian's original semantic network was a computer data structure that mimicked a dictionary. In the KL-ONE implementation of a semantic network, the IS-A relationship took center stage. Thus, an ontology is (visually) a network of nodes (boxes) and links (arrows) connecting the nodes. Figure 1 shows a tiny excerpt of an ontology.[2]

The basic unit of knowledge in a terminology is a concept. *Drug, Pneumonia, Antibiotic* and *Erythromycin* are concepts. For each concept various kinds of attributes may be specified, e.g., name, ID number, synonyms and other alphanumeric attributes. These attributes provide additional information about the given concepts. In Figure 1, Dosage is an attribute.

Concepts may also refer to other concepts by relationships. In Figure 1, relationships are shown as thin arrows. Relationships have labels attached, such as "is-prescribed-for" and are called "semantic relationships." Semantic Relationships are distinct from the

Figure 1. Example of Medical Terminology applied to the treatment of bacterial pneumonia



special purpose IS-A relationships (bold arrows), which form a specialization/generalization hierarchy. Thus, *Erythromycin* IS-A *Antibiotic* says that *Antibiotic* is more general than *Erythromycin*. In other words, all real-world instances of *Erythromycin* form a subset of all real-world instances of *Antibiotics*. Similarly, *Pneumonia* IS-A *Disease*. The IS-A relationship allows simple forms of reasoning with the concepts of the terminology. The most popular form of reasoning is inheritance. Thus, *Erythromycin* has the attribute Dosage, even though it is not shown in the figure. Dosage is inherited (automatically) along the IS-A link (against the direction of the arrow) from the parent concept *Antibiotic*.

Many ontologies also include *individuals* and/or support different forms of logic-based reasoning. Axioms may be attached to concepts in the taxonomy. The most popular logic formalism is called First Order Predicate Logic (FOPL). Unfortunately, FOPL is neither decidable nor tractable, which means that finding true results may be impossible or may take exponential amounts of time (e.g., millions of years). Thus, weaker logical formalisms have been invented. Indeed, many older and all modern members of the KL-ONE family itself were reconceived as "Description Logics" (Baader et al., 2003).

Description Logics maintain the basic KL-ONE structure and aim for well-defined tradeoffs between computability and expressiveness (Brachman & Levesque, 1984). Meanwhile improved First Order Logic (FOL) provers have also been developed. Tsarkov and Horrocks (2003) presented a comparison between DL and FOL systems. Given the long history of *building* ontologies it is somewhat surprising that *using* ontologies is still not a straightforward process, a phenomenon referred to as *knowledge use paradox* in (Geller et al., 2004).

MAIN FOCUS

Ontologies come in many different shapes and forms. An early categorization of ontologies can be found in (Noy & Friedman, 1997). They distinguished between ontologies based on generality, domain, size, formalism used, etc. John Bateman's comprehensive ontology portal[3] lists ontologies in linguistics, medicine, geography, translation, information reuse, business, general knowledge, engineering, software, etc. Ontologies may also be categorized into terminological versus axiomatized ontologies. Fensel (2004) provides a good, compact review of ontologies and how they will be useful in Electronic Commerce. An easy introduction to ontologies in Bioinformatics is (Baclawski & Niu, 2006). A comprehensive account of many aspects of ontologies is (Gomez-Perez et al., 2004).

One of the most widely used and successful lexical ontologies is WordNet (Fellbaum, 1998). It has inspired similar projects in other languages, such as GermaNet (Hamp and Feldweg, 1997). The CYC project also

deserves special mention (Lenat, 1995). Envisioned by Doug Lenat and Ed Feigenbaum at Stanford, CYC (from enCYClopedia) was a ten-year project (1984-1994) intended to codify once and for all human common sense knowledge into a large set of concepts and axioms. CYC had what no other AI project had had before it, namely a multi-million-dollar budget and a dedicated development staff. Nevertheless, CYC was not finished by the planned end of the project and is still being actively developed by CYCORP, Lenat's company.[4]

We will focus on three areas of special interest: Ontologies for the Semantic Web, Ontologies in Database Integration and Ontologies in Data Mining.

Ontologies and the Semantic Web

Tim Berners-Lee, the inventor of the World-Wide Web and the co-inventor of the Semantic Web (Berners-Lee et al., 2001) describes "the third basic component of the Semantic Web, collections of information called ontologies." Furthermore, Ontologies "can be used … to improve the accuracy of Web searches … More advanced applications will use ontologies to relate the information on a page to … inference rules." While historically ontologies had been represented in formats based on the LISP programming language, ontologies on the Web are written as RDF triples[5] which are themselves implemented in an XML-based formalism. OWL is currently a popular language for implementing Web Ontologies.[6] Editing of OWL is possible with the widely used Protégé tool.[7] Logic-based reasoners have been developed for OWL, such as RACER[8] and Pellet.[9] The implementation language of choice is typically Java. It should be noted that even a successful Semantic Web would (most likely) not be equivalent to a truly intelligent system.

Ontologies and Schema Integration

Ontologies have become popular in the field of databases to overcome difficulties in areas such as schema integration. When databases have to be merged, large database schemata need to be integrated, which is a difficult task. Much research has gone into automating schema matching (Rahm and Bernstein, 2001), but it is still an unresolved problem. Ontologies have been recognized as a tool for this purpose (Hakimpour & Geppert, 2001). For example, recognizing that two column names of two tables refer to the same real world property requires the kind of human knowledge that is often represented in synonym lists of ontologies.

Ontologies and Data Mining

A number of natural connections between ontologies and (data) mining have been explored. As the building of ontologies by hand is a difficult and error prone process, attempts have been made to automate the building of ontologies. Some of those attempts have been based on Web mining and/or text mining, e.g., Missikoff et al. (2003). At the opposite end of the spectrum are approaches that use ontologies to improve data or Web mining, e.g., Bernstein et al. (2005). A third connection between ontologies and data mining has been explored by Geller et al. (2005). In this approach, called *Raising*, raw transaction data is preprocessed using an ontology before the actual data mining step. As a result, rules with better support and more generality can be derived.

Medical Terminologies

One may argue that the most comprehensive ontologies (or ontology-like terminologies) today exist in the field of Medical Informatics. Research in Artificial Intelligence had concentrated on the properties of representation systems and reasoning systems, with building of actual (large) ontologies as a later goal, not realized until the start of the CYC project in 1984. Medical Informatics, in contrast, started as an attempt to organize the enormous amounts of existing medical terms, which go far beyond the mental capacity of any medical practitioner or even researcher, for practical purposes such as literature search.

The 1940 *Quarterly Cumulative Index Medicus* Subject Headings may be considered as the "grandfather" of modern medical terminologies.[10] This index was developed by the National Library of Medicine into the Medical Subject Headings (MeSH, 1963), which have been greatly extended and are still in use today. Today, the UMLS (Unified Medical Language System) is considered the most comprehensive and probably most important medical terminology. Under development since 1986 at the National Library of Medicine, the UMLS consists of over 100 other terminologies and its main part is therefore considered to be a Metathesaurus. The UMLS Semantic Network is

used as an upper layer for organizing the information in the Metathesaurus. Recently there has been great interest in the application of structural computer science techniques to the UMLS (Perl & Geller, 2003).

Another major medical terminology is SNOMED-CT (Systematized Nomenclature of Medicine - Clinical Terms). Contrary to the UMLS, which was the result of multiple integration processes, the SNOMED has been developed as a fairly homogeneous terminology by the College of American Pathologists (CAP). (The SNOMED CT is the result of the integration of the United Kingdom's National Health Service (NHS) Read codes with the prior SNOMED RT.).[11] The SNOMED CT contains over 311,000 concepts and more than 800,000 descriptions.[12] Since 2004, the SNOMED CT has been integrated into the UMLS.

Other important medical terminologies are ICD9-CM, HL7, and LOINC. In Europe, the GALEN project (Rogers et al., 2001), implemented in the GRAIL language, was developed for storing detailed clinical information about patients in a format useable by both medical doctors and computers. Due to the explosive growth of the field of genomics, the Gene Ontology (GO)[13] is also considered of increasing importance. Modern terminologies are expected to conform to Cimino's desiderata (Cimino et al., 1989), which include, e.g., Concept Orientation (non-redundant, non-ambiguous, permanent), multiple inheritance (called polyhierarchy by Cimino), and the use of Semantic Relationships.

FUTURE TRENDS

Semantic Web

The Semantic Web has made the first exciting steps out of the lab and into the commercial environment.[14] The Semantic Web provides a genuinely new mechanism of sharing islands of terminologies between different Web pages by referring to their URIs. Researchers involved in ontologies for the Semantic Web need to be careful to avoid some of the mistakes of their parent discipline Knowledge Representation, such as primacy of formal properties. A rapid expansion and wide scale sharing of RDF-encoded ontologies needs to be achieved. Writing of Semantic Web pages needs to become as easy as creating personal home pages, which is still a difficult goal. Comparing the speed of development of the World-Wide Web with the Semantic Web, we find that the WWW has both grown and gained acceptance much faster without any prior expectations.

Medical Terminologies

Medical Terminologies are increasingly included in production-quality systems ranging from hospitals to insurance companies. Because of that, dangers could arise for patient populations, unless great care is taken to ensure the correctness of all terms. This author believes that in the future *auditing of medical terminologies* will take center-stage. Some methods of auditing can be found in (Geller et al., 2003).

CONCLUSION

Ontologies and Medical Terminologies are major mechanisms for including human-like knowledge in software systems. As there is pressure on software developers to create more intelligent programs, the importance of ontologies and Medical Terminologies will only increase. However, in order to achieve widespread acceptance, great care has to be taken to make them both user-friendly and correct.

ACKNOWLEDGMENT

John Bateman has commented helpfully on a draft of this document.

REFERENCES

Baclawski, K. & Niu, T. (2006). *Ontologies for Bioinformatics*, Cambridge, MA: The MIT Press.

Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P. (Eds.) (2003). *The Description Logic Handbook: Theory, Implementation and Applications*, Cambridge, UK: Cambridge University Press.

Bemers-Lee, T., Hendler, J. & Lassila, O. (2001). The Semantic Web, *Scientific American* 284(5), 34-43.

Bernstein, A., Provost, F. J. & Hill, S. (2005). Toward Intelligent Assistance for a Data Mining Process: An Ontology-Based Approach for Cost-Sensitive

Classification. *IEEE Trans. Knowl. Data Eng*. 17(4), 503-518.

Brachman, R. J. & Levesque, H. J. (1984). The tractability of subsumption in framebased description languages, in *Proceedings of AAAI-84*, Austin, TX, 34-37.

Brachman, R. J & Schmolze, J. G. (1985). An overview of the KL-ONE knowledge representation system, *Cognitive Science*, 9(2), 171-216.

Cimino, J. J., Hripcsak G., Johnson S. B. & Clayton P. D. (1989). Designing an Introspective, Multipurpose, Controlled Medical Vocabulary, in: *Proceedings of the Thirteenth Annual Symposium on Computer Applications in Medical Care*, L. C. Kingsland (Ed.), New York, NY: IEEE Computer Society Press, pp. 513-518.

Fellbaum, C. (Ed.) (1998): *WordNet An Electronic Lexical Database*, Cambridge, MA: MIT Press.

Fensel, D. (2004). *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*, New York, NY: Springer Verlag.

Fikes R. and Kehler T. (1985). The role of frame-based representation in reasoning, *CACM* 28(9), 904-920.

Geller, J., Gu, H., Perl, Y. & Halper, M. (2003). Semantic refinement and error correction in large terminological knowledge bases, *Data & Knowledge Engineering*, 45(1), 1-32.

Geller, J., Perl, Y. & Lee, J. (2004). Guest Editors' introduction to the special issue on Ontologies: Ontology Challenges: A Thumbnail Historical Perspective, *Knowledge and Information Systems*, 6(4), pp. 375-379.

Geller, J., Zhou, X., Prathipati, K., Kanigiluppai, S. & Chen, X. (2005). Raising Data for Improved Support in Rule Mining: How to Raise and How Far to Raise, *Intelligent Data Analysis*, 9(4), 397-415.

Genesereth, M. R. (1991). Knowledge Interchange Format. *Principles of Knowledge Representation and Reasoning: Proceedings of the Second International Conference*, Cambridge, MA, Morgan Kaufmann Publishers, pp. 599-600.

Genesereth, M. R. & Nilsson, N. J. (1987). *Logical Foundations of Artificial Intelligence*. San Mateo, CA: Morgan Kaufmann Publishers.

Gomez-Perez, A., Fernandez-Lopez, M., & Corcho, O. (2004). *Ontological Engineering*, New York, NY: Springer Verlag.

Gruber, T. R., (1992). A translation approach to portable ontology specification, Knowledge Systems Laboratory, Technical Report KSL 92-71, Stanford University.

Gruber, T. R. (1993). A translation approach to portable ontologies, *Knowledge Acquisition*, 5(2), 199-220.

Hakimpour, F. & Geppert, A. (2001). Resolving semantic heterogeneity in schema integration: An ontology base approach, in *Proceedings of International conference on Formal Ontologies in Information Systems FOIS 2001*, Chris Welty and Barry Smith, (Eds.). New York, NY: ACM Press.

Hamp, B. & Feldweg, H. (1997): GermaNet - a Lexical-Semantic Net for German, in *Proceedings of ACL workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*", Madrid, 1997.

Humphreys, B. L., Lindberg, D. A. B,. Schoolman, H. M. , Barnett, G. O. (1998). The Unified Medical Language System: An Informatics Research Collaboration. *JAMIA* 5(1), 1-11.

Lenat, D. B. (1995), CYC: A Large-Scale Investment in Knowledge Infrastructure, *Communications of the ACM* 38(11). See also other articles in this special issue.

McCarthy, J. & Hayes, P. J. (1969). Some philosophical problems from the standpoint of artificial intelligence, in *Machine Intelligence 4*. B. Meltzer & D. Michie (Eds.), Edinburgh: Edinburgh University Press.

Minsky, M. (1975). A framework for representing knowledge, in *The Psychology of Computer Vision*, P. Winston, (Ed.), New York, NY: McGraw-Hill, pp. 211--277.

Missikoff, M., Velardi, P., Fabriani, P. (2003). Text Mining Techniques to Automatically Enrich a Domain Ontology. *Appl. Intelligence*, 18(3), 323-340.

Neches, R., Fikes, R. E., Finin, T., Gruber, T. R., Patil, R., Senator, T. & Swartout, W. R. (1991). Enabling Technologies for Knowledge Sharing, *AI Magazine* 12(3), 36-56.

Noy, N. F. & Hafner, C. D. (1997), The State of the Art in Ontology Design: A Survey and Comparative Review, *AI Magazine*, 18(3), 53-74.

Perl, Y. & Geller, J. (2003). Guest Editors' introduction to the special issue: Research on structural issues of the UMLS--past, present, and future, *Journal of Biomedical Informatics*, 36(6), 409-413. Special Issue: pp. 409-517.

Rahm, E. & Bernstein, P. A. (2001). A survey of approaches to automatic schema matching, *VLDB Journal*, 10, 334-350.

Quillian, M. (1968). Semantic Memory, in *Semantic Information Processing*, M. Minsky (Ed.), Cambridge, MA: MIT Press, pp. 227-270; reprinted in Collins & Smith (eds.), Readings in Cognitive Science.

Rogers J., Roberts, A., Solomon D., van der Haring, E., Wroe, C., Zanstra, P. & Rector A. (2001). GALEN ten years on: tasks and supporting tools. *Medinfo* 10(Pt 1): 256-260.

Tsarkov, D. & Horrocks, I (2003). DL reasoner vs. first-order prover. In *Proc. of the 2003 Description Logic Workshop (DL 2003)*, D. Calvanese, G. De Giacomo, E. Franconi (Eds.), Rome, Italy.

ENDNOTES

1 http://www.nlm.nih.gov/pubs/factsheets/umls-meta.html
2 We are using the meta-terminology of http://web.njit.edu/~geller/what_is_an_ontology.html in this chapter.
3 http://www.purl.org/net/ontoportal
4 http://www.cyc.com/
5 http://www.w3.org/RDF/
6 http://www.w3.org/2004/OWL/
7 http://protege.stanford.edu/
8 http://www.racer-systems.com/products/tools/protege.phtml
9 http://www.mindswap.org/2003/pellet/
10 http://www.nlm.nih.gov/mesh/intro_hist2006.html
11 http://www.ihtsdo.org/snomed-ct/snomed-ct0/
12 http://www.ihtsdo.org/snomed-ct/snomed-ct0/snomed-ct-components/
13 http://www.geneontology.org/
14 http://www.semantic-conference.com/

KEY TERMS

Description Logics: A description logic is an inheritance network, structurally similar to a semantic network or frame system. It differs from the above in that the meaning of each network structure can always be expressed by a translation into predicate logic. Description logics do not allow full logical reasoning but implement inheritance and classification reasoning (where to place a concept in a given inheritance network). The purpose of this limitation is to make reasoning tractable (computable in polynomial time).

Directed Acyclic Graph (DAG): The IS-A hierarchy of an ontology is either a tree structure or a Directed Acyclic Graph. In a tree every concept has exactly one parent (more general) concept that it is connected to. (The only exception is the root, which has no parent). In a DAG, a node might have several parents. However, when traversing the DAG from a node following the IS-A links in the direction of the given arrow heads, it is impossible to ever return to the start node. This implements the intuition that a concept cannot be more general and more specific than itself.

IS-A Hierarchy (Subclass Hierarchy, Concept Taxonomy): An IS-A hierarchy consists of concept nodes connected by IS-A links, forming a graph structure. IS-A links express generalization and are used for inheriting information downwards. Thus, a link from a concept LIVER to a concept ORGAN would express that ORGAN is a more general concept. At the same time, all information about the concept ORGAN would also apply to LIVER and is therefore said to be "inherited" by LIVER.

Medical Terminology: Most medical terminologies are collections of medical terms, which are structurally similar or sometimes even identical to ontologies. Historically, some medical terminologies existed in paper form before adequate computational resources became available.

Ontology: A representation of human-like knowledge in a computer-implemented graph structure. The major components of most ontologies are concepts, local information attached to concepts (attributes), IS-A

relationships and semantic relationships between pairs of concepts. Many ontologies also contain attached axioms for logical reasoning.

Semantic Relationship (Role): A semantic relationship expresses a real-world connection between two concepts, such as PERSON is-owner-of CAR, with is-owner-of being the relationship. The semantic relationships of an ontology together with the concepts form a graph in which cycles are not just allowed but the norm. In some systems semantic relationships appear in pairs, e.g., is-owner-of and is-owned-by, pointing into opposite directions.

Semantic Web: The Semantic Web is a new generation of the World-Wide Web (WWW), co-invented by Tim Berners-Lee, the inventor of the WWW. Its primary goal is to make the WWW machine-processable. A key ingredient of the Semantic Web is the use of ontologies, represented in a special format called RDF (Resource Description Framework).

UMLS: The UMLS (Unified Medical Language System) of the NLM (National Library of Medicine) is the largest existing medical terminology. It consists of the very large Metathesaurus, the Semantic Network, and the Specialist Lexicon. The Metathesaurus consists of over 100 medical terminologies. The Semantic Network consists of few (135) very general "semantic types" connected by 54 relationships. Every concept in the Metathesaurus has its meaning constrained by assigning it at least one, but often several, Semantic Types.

Order Preserving Data Mining

Ioannis N. Kouris
University of Patras, Greece

Christos H. Makris
University of Patras, Greece

Kostas E. Papoutsakis
University of Patras, Greece

INTRODUCTION

Data mining has emerged over the last decade as probably the most important application in databases. To reproduce one of the most popular but accurate definitions for data mining; "it is the process of nontrivial extraction of implicit, previously unknown and potentially useful information (such as rules, constraints and regularities) from massive databases" (Piatetsky-Shapiro & Frawley 1991). In practice data mining can be thought of as the "crystal ball" of businessmen, scientists, politicians and generally all kinds of people and professions wishing to get more insight on their field of interest and their data. Of course this "crystal ball" is based on a sound and broad scientific basis, using techniques borrowed from fields such as statistics, artificial intelligence, machine learning, mathematics and database research in general among others. Applications of data mining range from analyzing simple point of sales transactions and text documents to astronomical data and homeland security (*Data Mining and Homeland Security: An Overview*). Usually different applications may require different data mining techniques. The main kinds of techniques that are used in order to discover knowledge from a database are categorized into association rules mining, classification and clustering, with association rules being the most extensively and actively studied area. The problem of finding association rules can be formulated as follows: Given a large data base of item transactions, find all frequent itemsets, where a frequent itemset is one that occurs in at least a user-specified percentage of the data base. In other words find rules of the form X→Y, where X and Y are sets of items. A rule expresses the possibility that whenever we find a transaction that contains all items in X, then this transaction is likely to also contain all items in Y. Consequently X is called the body of the rule and Y the head. The validity and reliability of association rules is expressed usually by means of support and confidence. An example of such a rule is {smoking, no_workout→heart_disease (sup=50%, conf=90%)}, which means that 90% of the people that smoke and do not work out present heart problems, whereas 50% of all our people present all these together.

Nevertheless the prominent model for contemplating data in almost all circumstances has been a rather simplistic and crude one, making several concessions. More specifically objects inside the data, like for example items within transactions, have been attributed a Boolean hypostasis (i.e. they appear or not) with their ordering being considered of no interest because they are considered altogether as sets. Of course similar concessions are made in many other fields in order to come to a feasible solution (e.g. in mining data streams). Certainly there is a trade off between the actual depth and precision of knowledge that we wish to uncover from a database and the amount and complexity of data that we are capable of processing to reach that target.

In this work we concentrate on the possibility of taking into consideration and utilizing in some way the order of items within data. There are many areas in real world applications and systems that require data with temporal, spatial, spatiotemporal or ordered properties in general where their inherent sequential nature imposes the need for proper storage and processing. Such data include those collected from telecommunication systems, computer networks, wireless sensor networks, retail and logistics. There is a variety of interpretations that can be used to preserve data ordering in a sufficient way according to the intended system functionality.

BACKGROUND

Taking into consideration and using the order of items within transactions has been considered in another form and for an alternate goal; that is with the task of sequential pattern mining. Essentially in association rule mining we try to find frequent items that appear together in the same transactions. In sequential pattern mining we are interested in sets of items that appear frequently in different transactions. Thus association rules mining can be thought as an intra-transactional search process whereas sequential pattern mining as an inter-transactional.

A formulation of the task of sequential pattern mining is as follows (Dunham, 2003; Tan, Steinbach & Kumar, 2006): Let D be a database of customer transactions, and $I=\{i_1, i_2, \ldots, i_m\}$ be a set of distinct attributes called items. A transaction or else called an event is a non-empty collection of items (in most cases ordered), denoted as $(i_1, i_2, \ldots, i_k)$. A sequence α on the other hand is a collection of events such that $(\alpha_1 \rightarrow \alpha_2 \rightarrow \ldots \rightarrow \alpha_t)$, where α_i is an event. The support or frequency of a sequence, denoted as $\sigma(\alpha, D)$, is the total number of sequences in the database D that contain α. Consequently the process of sequential pattern mining concerns with extracting those sequential patterns whose support exceed a user predefined minimum support value; though the items within every transaction were reordered, thus destroying the initial purchase order.

Sequential pattern mining can be used in a wide range of areas from biology and medicine, to business processes and web sessions. The problem of mining sequential patterns was first introduced in (Agrawal and Srikant, 1995), where three algorithms were presented with algorithm AprioriAll having the best performance. An enhanced algorithm by the name GSP was proposed in (Srikant & Agrawal, 1996) that outperformed AprioriAll by up to 20 times. In the same period Mannila, Toivonen and Verkamo (1997) proposed a work for mining frequent episodes, which was further extended in (Mannila & Toivonen, 1996) in order to discover generalized episodes. Algorithms MEDD (Multi-Event Dependency Detection) and MSDD (Multi-Stream Dependency Detection) in (Oates, Schmill, Jensen, & Cohen, 1997) discover patterns in multiple event sequences, by exploring instead of the sequence space directly the rule space.

Other works include Zaki's SPADE (Zaki, 2001), a method that employs lattice based search techniques and simple joins in order to decompose the search space into sub-lattices small enough to be processed in main memory, thus reducing the number of database scans. PrefixSpan (Pei et al., 2001) employs an internal representation of the data made of database projections over sequence prefixes. Finally a family of works where time plays the dominant role are (Vautier, Cordier, & Quiniou, 2005) and the most recent work in (Giannotti, Nanni & Pedreschi, 2006). Interested reader can refer to (Zhao & Bhowmick, 2003) for a broader and more detailed view of the specific area.

ORDERED DATA: A NEW PERSPECTIVE

Traditional algorithms and approaches for association rules mining work on the rather naive assumption that every item inside a transaction has an influence on all the rest items inside the same transaction. Nevertheless items may appear together in several transactions but this does not necessarily mean they all influence one another. For example suppose we have a transaction that contains the following items in the exact order as they were purchased:

[V, G, A, S, C]

Then according to current approaches item V is considered to have influenced the purchase of all other items (i.e. items G, A, S and C). Items G, A, S, C are also considered to have influenced the purchase of all other items inside the transaction. However the reasonable question that arises from this assumption is how an event that occurred in the future could have influenced an event in the past. More specifically how can we claim for example that the choice of item A was influenced by the purchase of item S or C since when A was purchased items S and C were not even in the basket. It is fairly logical to presume that there exists the possibility that an item purchased might have an influence on the decision to purchase or not an item afterwards, but how can we claim the opposite? This scenario is especially evident in on-line stores where the purchases are made in a chain-like fashion (i.e. one item at a time and usually without following some predefined shopping list), resembling a lot the web browsing behaviour of the world wide web users. Hence in our opinion every item inside a transaction influences only the subsequent items inside the same transaction. This leads us to the following observation: the choice

of any item inside a transaction can be considered to be independent from the choice of all subsequent items inside the same transaction and is only influenced by preceding items.

Taking the order of items into consideration is not as trivial as it seems, and many questions and ambiguities would have to be solved. Bellow we mention some of the most important ones.

Order Recording

In order for the above observation to be in effect and to be able to be utilized, the initial purchase order will have to be preserved. As noted before though, traditional algorithms for association rules mining up to now do not preserve the order by which the items inside a transaction were purchased, but rather order them in various ways mainly so that they can facilitate more efficient counting (e.g. according to Agrawal and Srikant (1994) as well as in almost all works thereafter it is assumed that items in each itemset as well as in every transaction are lexically ordered so that the formation of the set of candidate itemsets and their counting can be done more effectively). Even with sequential pattern mining there exists also a reordering phase that destroys the initial purchase order.

Nevertheless current systems and databases could record the order of items with little modifications, requiring practically little effort and costs. Depending on the precision and the accuracy of rules we want to discover we could record only the order, the date or even the exact time that an item was chosen. Consequently we could propose the obvious rules such as people that buy some items with a specific order also buy some other, or people that buy these items then they should be proposed also some others after a specific time period. Again we must note that in comparison to sequential pattern mining, we discover ordered pairs in every transaction and not between transactions, and also we keep the initial ordering within every transaction.

Order Quantification

The first question that arises is how we are going to quantify the order of items within every transaction. Let's take for example the transaction discussed above (i.e. transaction [V, G, A, S, C]), assuming again that the initial order of appearance of every item has been preserved. As noted previously we assume that items influence only the choice of items that come after them (e.g. item A influences the choices of items S and C). So one obvious way to model this suggestion would be to suppose that every item has the same degree of influence on all the following items. An alternative suggestion though would be to take into consideration also the actual distance between any two items. For example item G has the biggest degree of influence on the item just after it (i.e. item A) and this influence diminishes as we move away from it. The degree of influence of every item could be coupled with the time factor associated with every purchase. For example if we chose item G and in the same transaction but after 30 whole minutes we chose item A, then the two purchases could be considered independent (of course this depends on the nature of items sold and the application). This adds some to the complexity of the problem but leads us to far more accurate suggestions. Also one might consider the case where in long transactions the influence of an item actually disappears after a certain number of items.

Assigning Importance to Every Item

All items are not equally important. If we take for example the case of a retail store, an expensive item or more accurately an item with large profit margins is probably more important than another one with fewer profit margins even if it does not appear that often. A classical example is the case of chewing gums and caviar, where despite the fact that chewing gums appear in almost all transactions if we wish to be accurate they cannot be considered more important than caviar. Also inherent differences in the natures of items or in their purchasing patterns suggest also different approaches for handling them, in terms of their importance. Another example would be that with items that are rare but on the other hand appear always together. For example people that work as lifeguards present an extremely high percentage of skin cancer, despite the fact that both these events when viewed separately are very rare. Consequently the importance of an item, regardless of how it would be measured, should be coupled efficiently with the quantification of items within transactions in order to assign the correct degree of influence to every item.

Cooping with Increased Complexity

Probably the most serious problem that would have to be surmounted regards the increased complexity that the above observations imply. As noted previously the more accurate rules we wish to produce and the more information we use, the more complex and sluggish a specific system becomes. So we have to either disregard some information or to find ways to reduce system complexity.

The Matrix Based Itemset Recommendation

In this section we briefly present a simple but straightforward solution for taking into consideration the order of items within transactions. Let I be the set of items. Let D be the set of transactions, where each transaction T is a set of items such that $T=\{i_1, i_2, i_3,\ldots, i_k\}$, where $i_i \in I$. The specific transaction is shown here as an ordered list of items since in our case items in a transaction are stored in the exact order as they first appear.

What we propose is a system that tries to mimic the function of an on-line collaborative filtering system that manages though also to take into consideration the exact order of the items in the transactions, as well as their special importance. Every item in our database is represented in a two dimensional table, called adjacency matrix like the one shown in Figure 1. Using every transaction we assign weights between all items using a scheme like the one proposed above (see section **Order Quantification**). For example suppose we have transaction [A, N, C, I, J, M]. Assigning a weight of 1 to every item preceding another one results in the adjacency matrix shown bellow. This process is repeated for every transaction, summing up the corresponding weights at every step.

Proposal Phase

After having constructed the adjacency matrix using every available transaction, our system is ready to make itemsets proposals to its users, using a very simple procedure. Suppose a customer puts an item in his basket. Then the system reads the entries for the specific item, sorts the values of all items present in descending order and proposes the most significant ones to the users. When two or more items are chosen then we read their lists, combine the value for every entry, sort them in descending order and again propose the most significant ones to the users.

FUTURE TRENDS

In the context of this work there exist two intriguing and important possible future directions. Using a time stamp for ordering items within transactions is an important attribute of each dataset since it can give us more accurate and useful information in the process of data mining. It would be very interesting though if we could couple the time attribute with a space attribute. In other words what would happen if we stored also the locations or the distances where some purchases were

Figure 1. An adjacency matrix

	A	C	F	I	J	M	N
A	0	1	0	1	1	1	1
C	0	0	0	1	1	1	0
F	0	0	0	0	0	0	0
I	0	0	0	0	1	1	0
J	0	0	0	0	0	1	0
M	0	0	0	0	0	0	0
N	0	1	0	1	1	1	0

made and process the whole database as a spatiotemporal dataset. Equally interesting would be the recording and utilization of items removal from transactions or data in general. Objects are not only inserted in a database but also removed. For example in a transaction some items are chosen and later removed. So it would be very interesting to examine how we could incorporate this information also.

CONCLUSION

In this work we tried to bring out and examine a parameter in data mining research that was left aside, namely the order of items within every transaction. Consequently we pinpointed other techniques that might have even vaguely or in some small extent engaged with a similar notion. We gave some possible problems that one might face trying to utilize items ordering and we concluded with the proposal of a possible solution to the specific problem. Although it may seem too simple, it has the advantage of being very straightforward and thus easy to grasp and implement.

Among other applications our proposed approach is especially suitable in the area of Web Mining, where the discovered rules could be used to extract frequent patterns based on the initial preserved order of the data in log files. For example for building a link predictor; i.e. a system-aided Web navigation application that processes and analyzes the click-stream-paths from many different users of a site over time and generates rules, which can be used to predict future navigational behavior of other users under a personalized framework. Also these rules could also be applied to enhance server performance through HTTP request prediction.

Researchers and techniques have to move away from the rather static way data were mostly treated until now especially regarding association rules, and assume a more dynamic view. Even in the most static database every day new data is inserted, old data is deleted while some existing is updated.

REFERENCES

Agrawal, R., & Srikant, R. (1995). Mining Sequential Patterns. In *Proceedings 11th IEEE ICDE Conference*, Taipei, Taiwan, March, (pp. 3-14).

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining generalized association rules. In *Proceedings 20th VLDB Conference*, (pp. 487-499). Santiago, Chile.

Castelo, R., Feelders, A. J., & Siebes, A. (2001). MAMBO: Discovering association rules based on conditional independencies. *LNCS 2189,* (pp. 289-298).

Data mining and homeland security: An overview. Congressional Research Service Reports on Intelligence and Related Topics. Retrieved January 28, 2006 from http://www.fas.org/sgp/crs/intel/RL31798.pdf

Dunham, M. (2003). *Data mining: Introductory and advanced topics*. Prentice Hall.

Giannotti, F., Nanni, M., & Pedreschi, D. (2006). Efficient mining of temporally annotated sequences.In *Proceedings 6th SIAM SDM*, (pp. 348-359). Bethesda, MD.

Mannila, H., Toivonen, H., & Verkamo, I. (1997). Discovering of frequent episodes in event sequences. *Data Mining and Knowledge Discovery Journal, 1(3),* 259-289.

Mannila, H., & Toivonen, H. (1996). Discovering generalized episodes using minimal occurrences. In *Proceedings 2nd KDD,* Portland, Oregon, USA, (pp. 146-151).

Oates, T., Schmill, M. D, Jensen, D., & Cohen, P. R. (1997). A family of algorithms for finding temporal structure in data. In *6th International Workshop on AI and Statistics,* (pp. 371-378). Fort Lauderdale, Florida.

Pei, J. et al. (2001). PrefixSpan: Mining sequential patterns by prefix-projected growth. In *Proceedings 17th ICDE*, (pp. 215-224). Heidelberg, Germany.

Piatetsky-Shapiro, G., & Frawley, W. J, (1991). *Knowledge discovery in databases.* AAAI/MIT Press.

Srikant, R., & Agrawal, R. (1996). Mining sequential patterns: Generalizations and performance improvements. In *Proceedings 5th EDBT*, (pp. 3-17). Avignon, France.

Tan, P., Steinbach, M., & Kumar, V. (2006). *Introduction to data mining.* Addison-Wesley.

Tanner, M. A. (1996). *Tools for statistical inference: Methods for the exploration of posterior distribu-*

tions and likelihood functions (3rd edition). Springer Verlag.

Vautier, A., Cordier, M.-O., & Quiniou R. (2005, July). An inductive database for mining temporal patterns in event sequences. In *Proceedings* 19th *IJCAI,* (pp. 1640-1641). Edinburgh, Scotland. July.

Zaki, M. J. (2001). SPADE: An efficient algorithm for mining frequent sequences. *Machine Learning Journal, 42(1/2),* 31-60.

Zhao, Q., & Bhowmick, S. S. (2003). *Sequential pattern mining: A survey*. Technical Report 2003118. Centre for Advanced Information Systems, School of Computer Engineering, Nanyang Technological University, Singapore.

KEY TERMS

Association Rules: An implication of the form $X \rightarrow Y$ where X and Y are itemsets. An association rule expresses the possibility that whenever we find a transaction that contains all items in *X*, then this transaction is likely to also contain all items in *Y*.

Data Mining: The nontrivial extraction of implicit, previously unknown, and potentially useful information from data.

Itemset: A set of database items/entities occurring together.

Item Reordering: The preprocessing phase, where items are reordered in order to facilitate more efficient counting.

Ordered Data: Data (itemsets) where their original order of appearance is preserved.

Order Preserving Data Mining: The task concerned with finding association rules among ordered data.

Sequential Pattern Mining: The task concerned with finding sets of items that appear frequently in different transactions.

Outlier Detection

Sharanjit Kaur
University of Delhi, India

INTRODUCTION

Knowledge discovery in databases (KDD) is a non-trivial process of detecting valid, novel, potentially useful and ultimately understandable patterns in data (Fayyad, Piatetsky-Shapiro, Smyth & Uthurusamy, 1996). In general KDD tasks can be classified into four categories i) Dependency detection, ii) Class identification, iii) Class description and iv) Outlier detection. The first three categories of tasks correspond to patterns that apply to many objects while the task (iv) focuses on a small fraction of data objects often called outliers (Han & Kamber, 2006). Typically, outliers are data points which deviate more than user expectation from the majority of points in a dataset.

There are two types of outliers: i) data points/objects with abnormally large errors and ii) data points/objects with normal errors but at far distance from its neighboring points (Maimon & Rokach, 2005). The former type may be the outcome of malfunctioning of data generator or due to errors while recording data, whereas latter is due to genuine data variation reflecting an unexpected trend in data. Outliers may be present in real life datasets because of several reasons including errors in capturing, storage and communication of data. Since outliers often interfere and obstruct the data mining process, they are considered to be nuisance.

In several commercial and scientific applications, a small set of objects representing some rare or unexpected events is often more interesting than the larger ones. Example applications in commercial domain include credit-card fraud detection, criminal activities in e-commerce, pharmaceutical research etc.. In scientific domain, unknown astronomical objects, unexpected values of vital parameters in patient analysis etc. manifest as exceptions in observed data. Outliers are required to be reported immediately to take appropriate action in applications like network intrusion, weather prediction etc., whereas in other applications like astronomy, further investigation of outliers may lead to discovery of new celestial objects. Thus exception/outlier handling is an important task in KDD and often leads to a more meaningful discovery (Breunig, Kriegel, Raymond & Sander, 2000).

In this article different approaches for outlier detection in static datasets are presented.

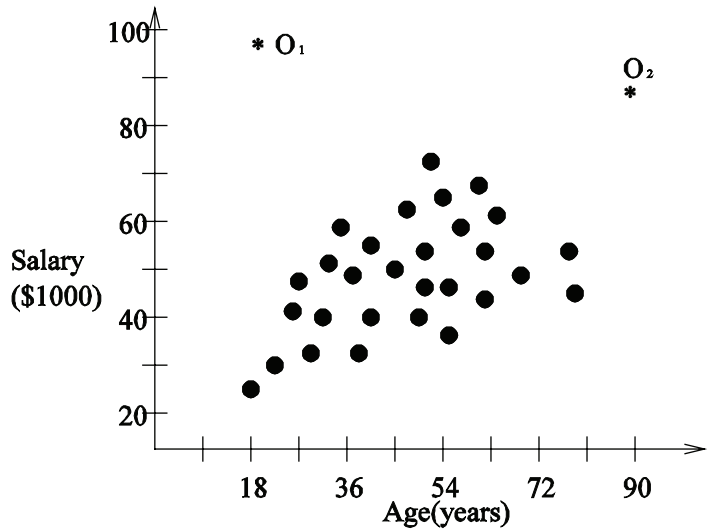
BACKGROUND

Outliers are data points which deviate *much* from the majority of points in a dataset. Figure 1 shows two outliers (O_1 and O_2) in Employee dataset with two attributes age and salary. Points O_1 and O_2 represent employees drawing high salary with age 18 and 90 respectively. These points are considered outliers because i) there is no other point in their neighborhood and ii) they are substantially different from the rest of points. Further exploration of such points may reveal some interesting facts.

Although exact definition of an outlier is application and context dependent, two commonly used general definitions for outliers are as follows. The classical definition is given by Hawkins (Hawkins, 1980) according to which, an outlier is an observation that deviates so much from other observations so as to arouse suspicion that it was generated by a different mechanism. A more recent definition, given by Johnson (Johnson, 1992), defines outlier as an observation which appears to be inconsistent with the remainder of the dataset.

Outlier detection in statistics has been studied extensively both for univariate and multivariate data, where a point not falling inside the distribution model is treated as outlier. Most of the approaches used in statistics are either suitable for univariate data or data with known distribution model e.g. Normal, Poisson etc. (Maimon & Rokach, 2005). In univariate approaches, only one feature is used, whereas in multivariate approaches multiple features are used to distinguish outliers from the normal objects. Multivariate approaches, which

Figure 1. Distinction between normal data and outliers



are computationally more expensive than univariate approaches, yield efficient results for low dimensional numeric dataset with known distribution model. However, for real life multidimensional datasets with unknown distribution, expensive tests for model fitting need to be performed. Data mining approaches do not assume any distribution model and overcome some of these limitations.

DATA MINING APPROACHES FOR OUTLIER DETECTION

Outlier detection is an important area in data mining to reveal interesting, unusual patterns in both static and dynamic datasets. In this article, we focus on non-parametric approaches for outlier detection in static datasets. These approaches detect outliers without any prior knowledge about underlying dataset and view dataset *S* as consisting of two components.

$$S = P + O \quad (1)$$

Here *P* represents normal data points and *O* represents exceptions or outliers. These approaches are categorized as: i) Clustering-based approach, ii) Distance-based approach and iii) Density-based approach, on the basis of their most distinct feature (Breunig, Kriegel, Raymond & Sander, 2000; Papadimitriou, Kitawaga, Gibbons & Faloutsos, 2003) .

Clustering-Based Approach

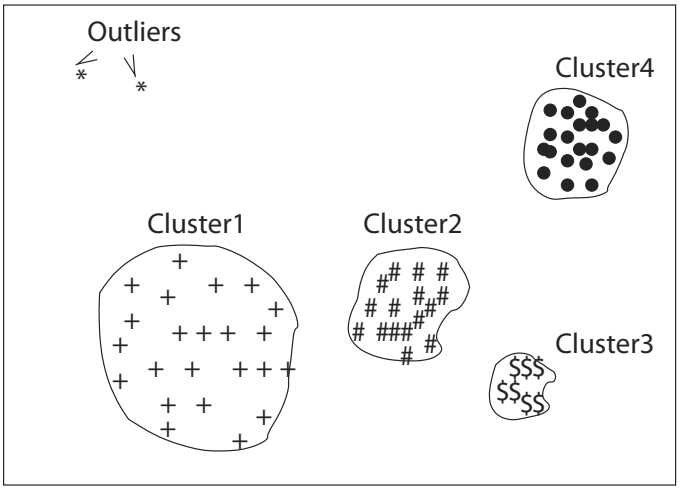
Clustering is a database segmentation technique which partitions a dataset into unknown groups as per prevalent data characteristics (Han & Kamber, 2006). Grouping is done such that data points in one group are more similar to each other than those belonging to different groups. These groups are called clusters. The data points which are not member of any cluster (Figure 2) are reported as outliers (Maimon & Rokach, 2005).

Clustering algorithms for large datasets like Balanced Iterative Reducing and Clustering using Hierarchy (BIRCH), Density Based Spatial Clustering of Applications with Noise (DBSCAN) and Clustering Using Representatives (CURE) report outliers as a by-product of the clustering process (Dunham, 2004). The objective of these algorithms, however, is to optimize clustering and not outlier detection.

BIRCH maintains compressed information required for clustering in *Cluster Feature Tree* (CF-Tree), which is a balanced tree. Each leaf in CF-Tree represents a set of points with a specified diameter and is treated as a pseudo point during clustering. The algorithm removes a leaf if the number of points is less than a user-defined threshold (λ) and reports all points as outliers.

DBSCAN uses a formal notion of density reachability to discover arbitrary shaped clusters with user defined minimum neighborhood δ and density *m*. *Density* of a point *p* is defined as a minimum number of points within a certain distance from *p*. A point is

Figure 2. Outliers detection using clustering approach



declared an outlier if there are less than *m* points in δ-neighborhood.

CURE uses both hierarchical and partitioning approaches for clustering. Initial clustering is performed on a sample drawn from the dataset. The sample is divided into equal sized partitions. Each partition is clustered using hierarchical approach. Clusters with points less than user specified threshold, are removed and points are reported as outliers.

Major disadvantage of clustering-based approach for outlier detection is that there is no standard measure to quantify *degree of outlyingness*. Outliers discovered are highly dependent on clustering scheme used and their detection criteria can not be inferred from the clustering method (Papadimitriou, Kitawaga, Gibbons & Faloutsos, 2003).

It is worth mentioning here that classical *k-Means* clustering algorithm does not differentiate between normal data and outliers. It reports clustering considering all data points including outliers, thereby deteriorating the quality of clustering scheme. Some refinements of *k-Means* have been suggested using rough set theory to improve its performance in the presence of outliers (Peters, 2005).

Distance-Based Approach

Distance-based approach for outlier detection computes neighbors of each point in the dataset using some distance metric like Euclidean distance or Mahalanobis distance. Although both metrics give comparable result but high computational complexity of Mahalanobis distance due to covariance matrix, makes it less popular as compared to Euclidean distance.

Knorr & Ng (1998) has proposed a distance-based method which overcomes limitations of clustering approach. The authors proposed the notion of distance-based outliers where *outlyingness* of a point w.r.t. another point is measured by Euclidean distance metric. For any two *d*-dimensional points *p* and *q*, *dist*(*p*,*q*) is given as follows:

$$dist(p,q) = \sqrt{\sum_{i=1}^{d} (p_i - q_i)^2} \quad (2)$$

In a dataset *S* of size *N*, an object *o* is defined as *DB*(*f*,*D*)-outlier if at least fraction *f* of objects in *S* lies at distance greater than *D* from *o*. For each object, neighborhood objects within distance *D* are counted and if the count is less than $N(1-f)$, the object is reported as an outlier. A major limitation of this approach is the setting of two crucial parameters *f* and *D*. A new user may find it difficult to set appropriately these parameters for successful outlier detection. Further, this approach does not rank outliers i.e. an outlier with very few neighbors and an outlier with more neighbors within distance *D* are not differentiated (Ramaswamy, Rastogi & Shim, 2000).

A variation of this approach is proposed by Ramaswamy et. al. (Ramaswamy, Rastogi & Shim, 2000), where the distance of k^{th} nearest neighbor of point *p* denoted as $D^k(p)$ is used to declare *p* as an

outlier. Given the number of nearest neighbors (k) to be checked and number of outliers to be reported (n), a point p is reported as outlier if no more than $n - 1$ points in the dataset have a higher value for $D^k(p)$. This approach reports n points with maximum $D^k(p)$ as *top-n* outliers. Although this approach ranks outliers without user specified distance D, it still requires two input parameters k and n and does not work efficiently for high dimensional datasets.

In order to overcome the computational expense of distance-based approach, Knorr & Ng propose following three strategies.

a. **Use of Index structure:** Various standard multidimensional indexing structures like k-d tree, R-tree etc. are used to execute a range search within distance D for each object o, to find its $N(1-f)$ neighbors. The worst case complexity is $O(dN^2)$ for d dimensions. Further additional cost of building index structure makes this strategy uncompetitive (Knorr & Ng, 1998). Maintaining minimum bounding rectangle (MBR) for each node optimizes the usage of index structure to reduce search time (Ramaswamy, Rastogi & Shim, 2000). While searching neighbors, if minimum distance between p and MBR of a node exceeds $D^k(p)$, then subtree of the node is not traversed.

b. **Use of Nested Loop:** It is a brute force approach for outlier detection where distance of each point is computed w.r.t rest of points in dataset. It is a computationally expensive approach with worst case complexity $O(N^2)$, but avoids the cost of building an index structure. To further optimize, Knorr & Ng (Knorr & Ng, 1998) proposed a variation by dividing dataset in blocks of fixed size. It reads two blocks at a time in the memory, referred as *first* and *second array*. For every point in *first array*, a count of neighboring points within distance D is maintained. As soon as count exceeds $N(1-f)$, it is marked as *non-outlier* and the process continues with the next point. Subsequently, the neighbors of unmarked points are searched in *second array* and count is updated. If points still remain unmarked, then *second array* is swapped by another unprocessed block and steps are repeated. Once all blocks are processed, each unmarked point in *first array* is reported as outlier. The process is repeated for the rest of data blocks which have not served as *first array* any time. Nested Loop is the most effective technique for detecting outliers from large datasets but is computationally more expensive.

c. **Use of Partition-based strategy:** This strategy partitions the data space such that points that are close together are assigned to the same partition. If a partition does not have any outlier then its points are not used in detecting outliers from rest of partitions. Knorr & Ng (1998) partition the dataset using cell-based approach which scales linearly with N, but is exponential w.r.t. d. Ramaswamy, Rastogi & Shim (2000) partition the dataset using some clustering method where each cluster represents a partition (p). It computes lower and upper bounds ($p.lower$, $p.upper$) for each partition p s.t. $p.lower \leq D^k(p) \leq p.upper$. A partition with $p.upper \leq mindist$ (minimum distance threshold) is pruned because it cannot contain outliers. Although this method works better than cell-based approach, it does not scale well for high dimensional dataset (Bay & Schwabacher, 2003).

Distance-Based Approach for High Dimensional Datasets

Major drawback of distance-based algorithms, explained so far, is their non-scalability with high-dimensional, large datasets. Orca and RBRP (Recursive Binning and Re-Projection) are the two recently proposed distance-based algorithms to detect *top-n* outliers from high-dimensional, large datasets.

ORCA (Bay & Schwabacher, 2003) optimizes nested loop strategy using randomization and pruning to reduce its worst case complexity from quadratic to linear in time. A set of points referred as *examples* is selected randomly from the dataset. An *outlier score* for each *example* is maintained on the basis of its distance from closest neighbor. The algorithm uses Euclidean distance for continuous features and Hamming distance for discrete features. An *example* is removed/pruned as soon as its outlier score becomes less than a cut-off threshold (χ). ORCA shows near linear scaling behavior only when the example set has a large number of outlying points and updates χ on the basis of outliers detected so far. Its complexity is near quadratic when *examples* are from mixed data distribution with lesser number of outlying points.

RBRP (Ghoting, Parthasarthy & Otey, 2006) uses two phase approach to reduce worst case complexity

of ORCA. In the first phase, dataset is recursively partitioned into equisized groups (bins). In the second phase extension of nested loop technique is used to find outliers. RBRP outperforms ORCA in scalability because of its pre-processing step for faster determination of approximate nearest neighbors.

Intuitively, distance-based approach works best for outliers possessing low neighborhood densities. As outlier is defined on the basis of some global parameters like *D* or *k*, distance-based approach does not give good result for the dataset with both sparse and dense regions (Tang, Chen, Fu & Cheung, 2006).

Density-Based Approach

Outlier detection algorithms using density-based approach compute outlying factor of each data point on the basis of number of points in the neighborhood (density). The main objective is to find how dense is the neighborhood of the point or how much it is alienated from others. LOF (Local Outlying Factor) and LOCI (Local Correlation Integral), two recent density-based algorithms are described below.

LOF (Breunig, Kriegel, Raymond & Sander, 2000) computes a *local outlier factor* (*lof*) for each point, indicating its *degree of outlyingnes*s. This *outlier factor* is computed using local density of the point and that of its neighbors. Neighborhood of a point is determined by the area covering a user specified minimum number of points (*k*). For a point *p*, the algorithm first computes *k*-nearest neighbor i.e. $D^k(p)$. Subsequently, for given *k*, it computes *reachability distance* for each point *p* w.r.t. another point *o* ($reach\text{-}dist_k(p)$) as the maximum of $dist(p,o)$ and $D^k(p)$. Figure 3 explains the concept for $k = 5$. $reach\text{-}dist_k(s)$ is simply a distance between points *s* and *r* as they are very far from each other. But points *q* and *r* are closer, hence $reach\text{-}dist_k(q)$ is equal to $D^k(r)$. For each *p*, *lof* is computed using $reach\text{-}dist_k(p)$ and those of *p*'s *k*-nearest neighbors. Points with high *lof* are reported as outliers. Although LOF works effectively for high-dimensional datasets, but selection of *k* is non-trivial.

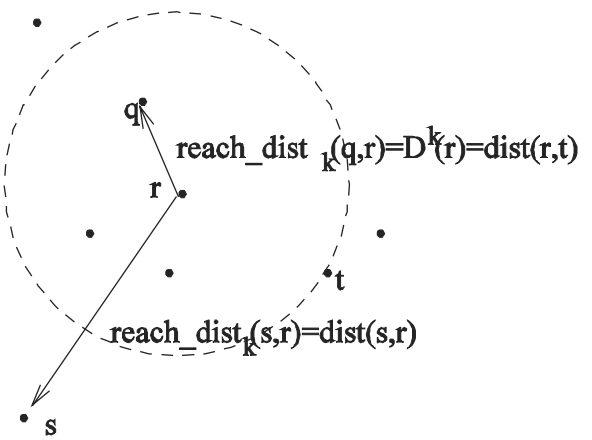
A similar technique LOCI (Papadimitriou, Kitawaga, Gibbons & Faloutsos, 2003), handles the difficulty of selecting values for *k* in LOF using *local correlation integral*. Instead of just computing *outlier score* for each point, it provides detailed information like the data in the vicinity of the point, diameter of neighboring area and inter-region distance etc.. It detects both isolated outliers as well as outlying clusters.

Density-based approach for outlier detection is more powerful than the distance-based approach when a dataset contains patterns with diverse characteristics like density, compactness etc. (Tang, Chen, Fu & Cheung, 2006). It does not suffer from the local density problem because only restricted neighbors of each data point are taken into account while computing *degree of outlyingness*. It also does not scale well for high-dimensional datasets because of index structure used for neighbor search.

OUTLIER DETECTION IN MIXED DATASET

Real life applications require both continuous as well categorical attributes, and detection of outliers from mixed data is still a challenging task. Ghoting, Otey & Parthasarathy (2004) propose LOADED (Link-based Outlier and Anomaly Detection in Evolving Datasets) which is probably the first algorithm to detect outliers from high-dimensional, evolving data with mixed attributes. The algorithm uses a link-based metric that allows outlier detection using efficient determination of dependencies between categorical and continuous attributes. Although it can be tuned to application needs (computation cost vs. accuracy), its performance degrades with increase in the number of categorical attributes.

Figure 3. Computation of $reach\_dist_k$ of points q and s w.r.t. point r for k=5 (Adapted from Breunig, Kreigel, Raymond, & Sander 2000)



FUTURE TRENDS

Outlier detection has been extensively researched for high-dimensional, large static datasets. With rapid change in data capturing technologies, it has become essential for many applications like network monitoring, video capturing, medical condition monitoring etc. to detect outliers on-line. Detection of such outliers is an upcoming and challenging area. Rough-set theory, Neural Network, Density estimation etc. are some upcoming techniques for on-line outlier detection. Different streaming window models like sliding window model, damped window model can be used for this purpose. Bhatnagar & Kaur (2007) detect outliers on the fly using grid-based clustering algorithm in stream.

CONCLUSION

The article introduces the problem of outlier detection and presents three data mining approaches for outlier detection in massive static datasets viz. clustering, distance and density computation. The basic concepts underlying each approach are discussed. Clustering-based approach reports outliers as a by product of clustering and do not rank outliers on the basis of their *outlyingness*. Density-based approaches perform better than distance-based approach for datasets with dense and sparse regions. Outlier detection in stream is an upcoming and challenging task.

REFERENCES

Bay, S., & Schwabacher, M. (2003). Mining distance-based outliers in near linear time with randomization and a simple pruning rule. *In Proceedings of 9th ACM SIGKDD international conference on Knowledge Discovery and Data Mining.*

Bhatnagar, V., & Kaur, S. (2007). Exclusive and Complete Clustering. *In Proceedings of international conference on Database and Expert System Application*, Germany.

Breunig, M., Kriegel, H., Raymond T. Ng., & Sander, J. (2000). LOF : Identifying Density-Based Local Outliers. *In Proceedings of ACM SIGMOD Conference on Management of Data. Texas, USA.*

Dunham, M. (2004). Data Mining: *Introductory and advanced topics*. Pearson Education, Singapore.

Fayyad, U. N., Piatetsky-Shapiro, G., Smyth, P., & Uthurusamy, R. (1996). *Advances in Knowledge Discovery and Data Mining*. MIT Press.

Ghoting, A., Otey, M.E., & Parthasarathy, S. (2004). LOADED: Link-based outlier and anomaly detection in evolving datasets. *In Proceedings of 4th IEEE International Conference on Data Mining*. (387-390).

Ghoting, A., Parthasarthy, S., & Otey, M. E. (2006). Fast Mining of Distance-based Outliers in High-dimensional Datasets. *In Proceedings of SIAM International Conference on Data Mining.*

Hawkins, D. (1980). *Identification of Outliers*, Chapman and Hall.

Han, J., & Kamber, M. (2006). *Data Mining – Concepts and Techniques*, Morgan and Kauffman.

Johnson, R. (1992). *Applied Multivariate Statistical Analysis*, Prentice Hall.

Knorr, E. M., & Ng, R. (1998). Algorithms for mining distance-based outliers in large datasets. *In Proceedings of International Conference on Very Large Databases.*

Maimon, O., & Rokach, L. (2005). *Data Mining and Knowledge Discovery Handbook*, Springer.

Papadimitriou, S., Kitawaga, H., Gibbons, P. B., & Faloutsos, C. (2003). LOCI: Fast Outlier Detection Using the Local Correlation Integral. *In Proceedings of 19th International Conference on Data Engineering.*

Peters, G. (2005). Outliers in Rough k-Means Clustering. *In Proceedings of 1st International Conference on Pattern Recognition and Machine Intelligence.*

Ramaswamy, S., Rastogi, R., & Shim, K. (2000). Efficient Algorithms for Mining Outliers from Large Datasets. *In Proceedings of ACM SIGMOD Conference on Management of Data. Texas, USA.*

Tang, J., Chen, Z., Fu, A. W., & Cheung, D. W. (2006). Capabilities of Outlier detection schemes in large datasets, framework and methodologies. *Journal of Knowledge Information Systems*, 45-84.

KEY TERMS

Clustering: The process of partitioning dataset S into clusters $C_1,\ldots..,C_k$ by finding similarities between data as per characteristics found.

$$S=\bigcup_{i=1}^{k}C_i \quad and \quad C_i \cap C_j = \varphi \quad for \quad i \neq j$$

Data Mining: Extraction of interesting, non-trivial, implicit, previously unknown and potentially useful patterns from large dataset.

Density: It is the number of points in the ε-neighborhood of a point where ε is user input.

Outlier: A point which is substantially different from rest of the points either on single attribute or has an unusual combination of values for more than one attributes.

Local Outlier Factor (*lof*): It represents *outlyingness* of a point *p* w.r.t to its *k* neighbors and is computed as the average of ratio of the local reachability density of *p* and those of *p*'s *k*-nearest neighbors.

$$lof_k(p)=\frac{\sum_{o\in N_k(p)}\frac{lrd_k(o)}{lrd_k(p)}}{|N_k(p)|}$$

Local Reachability Density: It is used to measure compactness of a point *p* w.r.t to its *k* neighbors where highly compact indicates more reachable. It is computed as the inverse of the average reachability distance based on the *k*-nearest neighbors of point *p* .

$$lrd_k(p)=1/\left(\frac{\sum_{o\in N_k(p)} reach-dist_k(p,o)}{|N_k(p)|}\right)$$

where *lrd* stands for local reachability densityand $N_k(p)$ is number of points within $D^k(p)$

Reachability Distance: The *reachabilty distance* of point *p* w.r.t point *o* is defined as

$reach\text{-}dist_k(p,o) = \max\{D^k(o), dist(p,o)\}$

where $D^k(o)$ is the distance of nearest k^{th} neighbor from *o* and *dist*(*p*,*o*) is the distance between points *p* and *o*.

Outlier Detection Techniques for Data Mining

Fabrizio Angiulli
University of Calabria, Italy

INTRODUCTION

Data mining techniques can be grouped in four main categories: clustering, classification, dependency detection, and outlier detection. Clustering is the process of partitioning a set of objects into homogeneous groups, or clusters. Classification is the task of assigning objects to one of several predefined categories. Dependency detection searches for pairs of attribute sets which exhibit some degree of correlation in the data set at hand.

The outlier detection task can be defined as follows: "Given a set of data points or objects, find the objects that are considerably dissimilar, exceptional or inconsistent with respect to the remaining data". These exceptional objects as also referred to as outliers.

Most of the early methods for outlier identification have been developed in the field of statistics (Hawkins, 1980; Barnett & Lewis, 1994). Hawkins' definition of outlier clarifies the approach: "An outlier is an observation that deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism". Indeed, statistical techniques assume that the given data set has a distribution model. Outliers are those points that satisfy a discordancy test, that is, that are significantly far from what would be their expected position given the hypothesized distribution.

Many clustering, classification and dependency detection methods produce outliers as a by-product of their main task. For example, in classification, mislabeled objects are considered outliers and thus they are removed from the training set to improve the accuracy of the resulting classifier, while in clustering, objects that do not strongly belong to any cluster are considered outliers. Nevertheless, it must be said that searching for outliers through techniques specifically designed for tasks different from outlier detection could not be advantageous. As an example, clusters can be distorted by outliers and, thus, the quality of the outliers returned is affected by their presence. Moreover, other than returning a solution of higher quality, outlier detection algorithms can be vastly more efficient than non ad-hoc algorithms.

While in many contexts outliers are considered as noise that must be eliminated, as pointed out elsewhere, "one person's noise could be another person's signal", and thus outliers themselves can be of great interest. Outlier mining is used in telecom or credit card frauds to detect the atypical usage of telecom services or credit cards, in intrusion detection for detecting unauthorized accesses, in medical analysis to test abnormal reactions to new medical therapies, in marketing and customer segmentations to identify customers spending much more or much less than average customer, in surveillance systems, in data cleaning, and in many other fields.

BACKGROUND

Approaches to outlier detection can be classified in supervised, semi-supervised, and unsupervised.

Supervised methods exploit the availability of a labeled data set, containing observations already labeled as normal and abnormal, in order to build a model of the normal class. Since usually normal observations are the great majority, these data sets are unbalanced and specific classification techniques must be designed to deal with the presence of rare classes (Chawla et al., 2004).

Semi-supervised methods assume that only normal examples are given. The goal is to find a description of the data, that is a rule partitioning the object space into an accepting region, containing the normal objects, and a rejecting region, containing all the other objects. These methods are also called one-class classifiers or domain description techniques, and they are related to novelty detection since the domain description is used to identify objects significantly deviating form the training examples.

Unsupervised methods search for outliers in an unlabelled data set by assigning to each object a score which reflects its degree of abnormality. Scores are usually computed by comparing each object with objects belonging to its neighborhood.

Data mining researchers have largely focused on unsupervised approaches. Most of the unsupervised approaches proposed in the data mining literature can be classified as deviation-based (Arning et al., 1996), distance-based (Knorr & Ng, 1998), density-based (Breunig et al., 2000), and MDEF-based (Papadimitriou et al., 2003).

Deviation-based techniques (Arning et al., 1996) identify as exceptional the subset I_x of the overall data set I whose removal maximizes the similarity among the objects in $I–I_x$.

Distance-based outlier detection has been introduced by (Knorr & Ng, 1998) to overcome the limitations of statistical methods: an object O is an outlier in a data set with respect to parameters k and R if at least k objects in the data set lie within distance R from O. This definition generalizes the definition of outlier in statistics. Moreover, it is suitable in situations when the data set does not fit any standard distribution.

(Ramaswamy et al., 2000), in order to provide a ranking of the outliers, modified the previous definition as follows: given two integers k and n, an object O is said to be the n-th top outlier if exactly $n–1$ objects have higher value for D^k than O, where D^k denotes the distance of the k-th nearest neighbor of the object.

Subsequently, (Angiulli & Pizzuti, 2002) with the aim of taking into account the whole neighborhood of the objects, proposed to rank them on the basis of the average distance from their k nearest neighbors, also called weight.

Density-based methods, introduced in (Breunig et al., 2000), are based on the notion of local outlier. Informally, the Local Outlier Factor (LOF) measures the degree of an object to be an outlier by comparing the density in its neighborhood with the average density in the neighborhood of its neighbors. The density of an object is related to the distance to its k-th nearest neighbor. Density-based methods are useful when the data set is composed of subpopulations with markedly different characteristics.

The multi-granularity deviation factor (MDEF), introduced in (Papadimitriou et al., 2003), is in principle similar to the LOF score, but the neighborhood of an object consists of the objects within an user-provided radius and the density of an object is defined on the basis of the number of objects lying in its neighborhood.

In order to discover outliers in spatial data sets, specific definitions are needed: spatial outliers (Shekhar et al., 2003) are spatially referenced object whose non-spatial attribute values are significantly different from the values of their spatial neighborhood, e.g. a new house in an old neighbourhood.

MAIN FOCUS

We overview several recent data mining techniques for outlier detection. We will focus on distance and density-based methods for large data sets, on subspace outlier mining approaches, and on algorithms for data streams.

Algorithms for Distance- and Density-Based Outliers

Distance-based outlier scores are monotonic non-increasing with respect to the portion of the data set already explored. This property allows to design effective pruning rules and very efficient algorithms.

The first two algorithms for mining distance-based outliers in large data sets were presented in (Knorr et al., 2000). The first one is a block nested loop that runs in time quadratic in the size of the data set. The second one is a cell-based algorithm whose temporal cost is exponential in the dimensionality of the data. These methods do not scale well for both large and high-dimensional data. Thus, efforts for developing scalable algorithms have been subsequently made. (Ramaswamy et al., 2000) present two algorithms to detect the top outliers. The first assumes that the dataset is stored in a spatial index. Since the first method is computationally expensive, they also introduce a clustering-based based algorithm, which has been tested up to ten dimensions.

The distance-based outlier detection algorithm ORCA (Bay & Schwabacher, 2003) enhances the naïve block nested loop algorithm with a simple pruning rule and randomization and exhibits good scaling behavior on large and high dimensional data. ORCA manages disk resident data sets and employs a memory buffer of fixed size. Its I/O cost, that is the total number of disk blocks accesses, my become quadratic due to the nested loop strategy. (Ghoting et al., 2006) use a divisive hierarchical clustering algorithm to partition data in a suitable data structure and then employ the strategy of ORCA on this structure, obtaining improved performances. Differently than ORCA, the whole data set must be accommodated in a main memory resident

ad-hoc data structure, hence it is not profitable if the data set does not fit in main memory. (Tao et al., 2006) point out that ORCA may present quadratic I/O cost in terms of pages read from disk. Then, they introduce an algorithm, named SNIF, whose goal is to reduce I/O cost. The algorithm loads in main memory a randomly selected subset of the data set, in order to prune inliers, and then scans three times the data set file. It can be used only with data belonging to a metric space. The DOLPHIN algorithm (Angiulli & Fassetti, 2007a) detects distance-based outliers in disk resident data sets with two scans. It maintains in main memory a provable small fraction of the data set and integrates pruning rules and database indexing technologies. This method is very fast and able to manage large collections of data. It is profitable for metric data, but can be used with any type of data for which a distance function is defined.

The HilOut algorithm (Angiulli & Pizzuti, 2005) detects the top distance-based outliers, according to the weight score, in a numerical data sets. It makes use of the Hilbert space-filling curve and consists of two phases: the first phase guarantees an approximate solution with temporal cost linear in the data set size, but, depending on the characteristics of the data set, may return also the exact solution; the second phase provides the exact solution. HilOut has been experimented with data up to 128 dimensions. It is the only distance-based outlier algorithm guaranteeing an approximate solution within a deterministic factor in time linear with respect to the data set size.

In domains where distance computations are very expensive, e.g. the edit distance between subsequences or the quadratic distance between image color histograms, determining the exact solution may become prohibitive. (Wu & Germaine, 2006) consider this scenario and describe a sampling algorithm for detecting distance-based outliers with probabilistic accuracy guarantees.

There are less proposals of scalable algorithms for the density-based definition than for the distance-based one. Among the algorithms for density-based outliers there are aLOCI (Papadimitriou et al., 2003), which detects MDEF-based outliers, and the micro-cluster-based algorithm of (Jin et al., 2001) for mining the top local outliers according to the LOF definition. The last method compresses the data set into hyper-spherical micro-clusters, and then computes upper and lower bounds for the LOF of objects in each micro-cluster and exploits them to prune inliers.

Distance and density based definitions are mainly considered in the context of unsupervised outlier detection, but can also be used as semi-supervised methods (Lazarevic et al., 2003). (Angiulli et al., 2006) introduce the concept of outlier detection solving set for the definition provided in (Angiulli & Pizzuti, 2005), which is a compressed representation for outlier prediction. (Angiulli, 2007) generalizes the aforementioned approach in order to encompass the definition provided in (Knorr & Ng, 1998) and compares it with well-established one-class classification methods.

(Xiong et al., 2006), besides presenting a novel method called HCleaner, experiment distance- and density-based outlier detection definitions as noise removal techniques, concluding that outlier detection can enhance quality of data analysis.

Subspace Outlier Mining

Aforementioned methods considered search for objects which can be regarded as anomalous by looking at the full dimensional space. In many real situations an object exhibits exceptional characteristics only when the attention is restricted to a subset of the features, also said subspace. Thus, some recent approaches to outlier detection deal with the enormous search space composed by all the subsets of the overall set of attributes.

(Knorr & Ng, 1999) focus on the identification of the intensional knowledge associated with distance-based outliers. First, they detect the outliers in the full attribute space and then, for each outlier O, they search for the subspace that better explains why it is exceptional, that is the minimal subspace in which O is still an outlier.

In many situations it is of interest to single out the properties that mostly deviate a single individual, which is not necessarily an outlier in the full dimensional space, from the whole population. The algorithm HighDOD (Zhang & Wang, 2006) searches for the subspaces in which an input point is an outlier according to definition provided in (Angiulli & Pizzuti, 2002). The algorithm is a dynamic subspace search method that utilizes a sampling-based learning process to identify promising subspaces. (Zhang et al., 2006) present a variant of the previously described task based on the definition provided in (Ramaswamy et al., 2000).

(Aggarwal & Yu, 2001) detect anomalies in a d-dimensional data set by identifying abnormal lower

dimensional projections. Each attribute is divided into equi-depth ranges, and then k-dimensional hyper-cubes ($k<d$) in which the density is significantly lower than expected are searched for. The technique is based on evolutionary algorithms. Also (Wei et al., 2003) exploit subspaces in order to detect outliers in categorical data sets. They search for sets of attributes being able to single out a portion of the data set in which the value assumed by some objects on a single additional attribute becomes infrequent with respect to the mean frequency of the values in the domain of that attribute.

Subspaces can be useful to perform example-based outlier detection: a set of example outliers is given as input and the goal is to detect the data set objects which exhibit the same exceptional characteristics as the example outliers. (Zhu et al., 2005) accomplish this task by means of a genetic algorithm searching for the subspace minimizing the density associated with hyper-cubes containing user examples.

Another interesting direction is exploiting subspaces to improve accuracy. In this context (Lazarevic & Kumar, 2005) experiment feature bagging: the same outlier detection algorithm is executed multiple times, each time considering a randomly selected set of features from the original feature set, and finally outlier scores are combined in order to find outliers of better quality.

Online and Data Streams Algorithms

Methods previously discussed are designed to work in the batch framework, that is under the assumption that the whole data set is stored in either secondary or main memory, and that multiple passes over the data can be accomplished. Hence these methods are not specific for the online paradigm or for processing data streams.

The SmartSifter system (Yamanishi, 2000) is an example of online algorithm. It employs an online discounting learning algorithm to learn a probabilistic model representing the data source. Every time a datum is input, SmartSifter evaluates how large the datum is deviated relative to a normal pattern.

In the data stream scenario, (Aggarwal, 2005) focuses on detecting rare events. Rare events are defined on the basis of their deviation from expected values computed on historical trends. (Subramaniam et al., 2006) present a distributed framework to approximate data distributions coming from a sensor network. Kernel estimators are exploited to approximate data distributions in order to report distance-based or MDEF-based outlying values in the union of the readings coming from multiple sensors. The STORM algorithm (Angiulli & Fassetti, 2007b) mines exact or approximate distance-based outliers under the sliding window model, where outlier queries are performed in order to detect anomalies in the current window.

FUTURE TRENDS

Searching for outliers in subspaces, or exploiting subspaces to improve accuracy or to explain outlierness is a relatively novel research topic, which has not been sufficiently explored. Also fast approximate unsupervised methods and condensed representations for the semi-supervised setting deserve further investigation. Currently there is great interest in novel definitions and methods specifically designed for rich data types, such has various forms of sequences and graphs, and for specific environments, such as grids and sensor networks. For instance, (Sun et al., 2006) present an approach to mine for sequential outliers using Probabilistic Suffix Trees, while (Palatin et al., 2006) consider a grid monitoring system which automatically detects oddly behaving machines by means of a distributed version of the HilOut algorithm. A very interesting direction concerns also the capability of exploiting domain knowledge in order to guide search for anomalous observations, as, for example, accounted for in (Angiulli et al., 2008).

CONCLUSION

We introduced the outlier detection task from a data mining perspective. We provided an up-to-date view on distance- and density-based methods for large data sets, on subspace outlier mining approaches, and considered also some algorithms for processing data streams. Throughout the work we presented different outlier mining tasks, pointed out peculiarities of the various methods, and addressed relationships among them. There is an intense activity in this field, which is even expected to increase due to the broad applicability of outlier detection techniques, and to challenging future directions.

REFERENCES

Aggarwal, C.C., & Yu, P. S. (2001). Outlier Detection for High Dimensional Data. *ACM International Conference on Managment of Data* SIGMOD, 37-46.

Aggarwal, C.C. (2005). On abnormality detection in spuriously populated data streams. *SIAM International Conference on Data Mining* SDM.

Angiulli, F., & Pizzuti, C. (2002). Fast Outlier Detection in High-Dimensional Spaces, *European Conference on Principles and Practice of Knowledge Discovery in Databases* PKDD, 15-26.

Angiulli, F., & Pizzuti, C. (2005). Outlier Mining in Large High-Dimensional Data Sets. *IEEE Transactions on Knowledge and Data Engineering* TKDE, 17(2), 203-215.

Angiulli, F., Basta, S., & Pizzuti, C. (2006). Distance-Based Detection and Prediction of Outliers. *IEEE Transactions on Knowledge and Data Engineering* TKDE, 18(2), 145-160.

Angiulli, F., & Fassetti, F. (2007a). Very Efficient Mining of Distance-Based Outliers. *ACM Conference on Information and Knowledge Management* CIKM, 791-800.

Angiulli, F., & Fassetti, F. (2007b). Detecting Distance-Based Outliers in Streams of Data. *ACM Conference on Information and Knowledge Management* CIKM, 811-820.

Angiulli, F. (2007). Condensed Nearest Neighbor Data Domain Description. *IEEE Transactions on Pattern Analysis and Machine Intelligence* TPAMI, 29(10), 1746-1758.

Angiulli, F., Greco, G., & Palopoli, L. (2008). Outlier Detection by Logic Programming. *ACM Transactions on Computational Logic* TOCL, forthcoming.

Arning, A., Aggarwal, R., & Raghavan, P. (1996), A Linear Method for Deviation Detection in Large Databases, *ACM International Conference on Knowledge Discovery and Data Mining* KDD, 164-169.

Bay, S. D., & Schwabacher, M. (2003). Mining Distance-Based Outliers in Near Linear Time with Randomization and a Simple Pruning Rule. *ACM International Conference on Knowledge Discovery and Data Mining* KDD, 29-38.

Barnett, V., & Lewis, T. (1994). *Outliers in Statistical Data*. John Wiley & Sons.

Breunig, M. M., Kriegel, H., Ng, R. T., & Sander, J. (2000). LOF: Identifying Density-based Local Outliers, *ACM International Conference on Management of Data* SIGMOD, 93-104.

Chawla, N. V., Japkowicz, N., & Kotcz, A. (2004). Editorial: Special Issue on Learning from Imbalanced Data Sets. *SIGKDD Explorations*, 6(1), 1-6.

Ghoting, A., Parthasarathy, S., & Otey, M. E. (2006). Fast Mining of Distance-Based Outliers in High Dimensional Datasets. *SIAM International Conference on Data Mining* SDM, 608-612.

Hawkins, D. (1980). Identification of Outliers. *Chapman and Hall*, London.

Jin, W., Tung, A. K. H., & Han, J. (2001). Mining Top-n Local Outliers in Large Databases. *ACM International Conference on Knowledge Discovery and Data Mining* KDD, 293-298.

Knorr, E. M., & Ng, R. T. (1998). Algorithms for Mining Distance-Based Outliers in Large Datasets, *International Conference on Very Large Databases* VLDB, 392-403.

Knorr, E. M., & Ng, R. T. (1999). Finding Intensional Knowledge of Distance-Based Outliers, *International Conference on Very Large Databases* VLDB, 211-222.

Knorr, E., Ng, R. T., & Tucakov, V. (2000). Distance-Based Outlier: Algorithms and Applications, *VLDB Journal*, 8(3-4):237-253.

Lazarevic, A., Ertöz, L., Kumar, V, Ozgur, A., & Srivastava, J. (2003). A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection. *SIAM International Conference on Data Mining SDM*.

Lazarevic, A., & Kumar, V. (2005). Feature Bagging for Outlier Detection. *International Conference on Knowledge Discovery and Data Mining* KDD, 157-166.

Palatin, N., Leizarowitz, A., & Schuster, A. (2006). Mining for Misconfigured Machines in Grid Systems. *ACM International Conference on Knowledge Discovery and Data Mining* KDD, 687-692.

Papadimitriou, S., Kitagawa, H., Gibbons, B., & Faloutsos, C. (2003). LOCI: Fast Outlier Detection Using the

Local Correlation Integral. *International Conference on Data Engineering* ICDE, 315-326.

Ramaswamy, S., Rastogi, R., & Shim, K. (2000). Efficient Algorithms for Mining Outliers from Large Data Sets, *ACM International Conference on Management of Data* SIGMOD, 427-438.

Shekhar, S., Lu, C.-T., & Zhang, P. (2003). A Unified Approach to Detecting Spatial Outliers. *GeoInformatica*, 7(2), 139-166.

Subramaniam, S., Palpanas, T., Papadopoulos, D., Kalogeraki, V., & Gunopulos., D. (2006). Online outlier detection in sensor data using non-parametric models. *International Conference on Very Large Databases* VLDB, 187-198.

Sun, P., Chawla, S., & Arunasalam, B. (2006). Mining for Outliers in Sequential Databases, *SIAM International Conference on Data Mining*, 94-105.

Tao, Y., Xiao, X., & Zhou, S. (2006). Mining Distance-Based Outliers form Large Databases in Any Metric Space. *ACM International Conference on Knowledge Discovery and Data Mining* KDD, 394-403.

Wei, L., Qian, W., Zhou, A., Jin, W., & Yu, J. X. (2003). HOT: Hypergraph-Based Outlier Test for Categorical Data. *Pacific-Asia Conference on Knowledge Discovery and Data Mining* PAKDD, 399-410.

Wu, M., & Jermaine, C. (2006). Outlier Detection by Sampling with Accuracy Guarantees. *International Conference on Knowledge Discovery and Data Mining* KDD, 767-772.

Xiong, H., Pandey, G., Steinbach, & M., Kumar, V. (2006). Enhancing Data Analysis with Noise Removal, *IEEE Transactions on Knowledge and Data Engineering* TKDE, 304-319.

Yamanishi, K., Takeuchi, J., Williams, G. J., & Milne, P. (2000). On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms, *ACM International Conference on Knowledge Discovery and Data Mining* KDD, 320-324.

Zhang, J., & Wang, H. (2006). Detecting Outlying Subspaces for High-Dimensional Data: the New Task, Algorithms, and Performance. *Knowledge and Information Systems*, 10(3), 333-355.

Zhang, J., Gao, Q., & Wang, H. (2006). A Novel Method for Detecting Outlying Subspaces in High-dimensional Databases Using Genetic Algorithm, *IEEE International Conference on Data Mining* ICDM, 731-740.

Zhu, C., Kitagawa, H., & Faloutsos, C. (2005). Example-Based Robust Outlier Detection in High Dimensional Datasets, *IEEE International Conference on Data Mining* ICDM, 829-832.

KEY TERMS

Distance-Based Outlier: An observation lying in a scarcely populated region of the feature space. Formally, an object such that at lest a fraction p of the objects lies greater than distance R from it, where p and R are user-provided parameters. This definition generalizes some statistical definitions for outliers.

Example-Based Outlier: An object whose exceptional characteristics are similar to those of a set of user-provided examples.

Local Outlier: An object whose density significantly deviates from the average density of its nearest neighbors. The density of an object can be defined in several ways; e.g., the reciprocal of the average distance to its nearest neighbors. A local outlier is also called a density-based outlier.

Noise: An object whose presence may degrade the quality of the output of a data analysis technique, as for example the predictive accuracy of a classification method.

Outlier: An observation which appears to be inconsistent with the remainder of the set of data.

Spatial Outlier: A spatially referenced object (e.g., houses, roads, traffic sensors, etc) whose non-spatial attribute values (e.g., age, owner, manufacturer, measurement readings, etc) are significantly different from the values of its spatial neighborhood.

Subspace Outlier: An object which becomes an outlier only when a specific subset of the overall set of attributes is considered.

Path Mining and Process Mining for Workflow Management Systems

Jorge Cardoso
SAP AG, Germany

W.M.P. van der Aalst
Eindhoven University of Technology, The Netherlands

INTRODUCTION

Business process management systems (Smith and Fingar 2003) provide a fundamental infrastructure to define and manage business processes and workflows. These systems are often called process aware information systems (Dumas, Aalst et al. 2005) since they coordinate the automation of interconnected tasks. Well-known systems include Tibco, WebSphere MQ Workflow, FileNet, COSA, etc. Other types of systems, such as ERP, CRM, SCM, and B2B, are also driven by explicit process models and are configured on the basis of a workflow model specifying the order in which tasks need to be executed.

When process models or workflows are executed, the underlying management system generates data describing the activities being carried out which is stored in a log file. This log of data can be used to discover and extract knowledge about the execution and structure of processes. The goal of process mining is to extract information about processes from logs.

When observing recent developments with respect to *process aware information systems* (Dumas, Aalst et al. 2005) three trends can be identified. First of all, workflow technology is being embedded in service oriented architectures. Second, there is a trend towards providing more flexibility. It is obvious that in the end business processes interface with people. Traditional workflow solutions expect the people to adapt to the system. However, it is clear that in many situations this is not acceptable. Therefore, systems are becoming more flexible and adaptable. The third trend is the omnipresence of event logs in today's systems. Current systems ranging from cross-organizational systems to embedded systems provide detailed event logs. In a service oriented architecture events can be monitored in various ways. Moreover, physical devices start to record events. Already today many professional systems (X-ray machines, wafer stepper, high-end copiers, etc.) are connected to the internet. For example, Philips Medical Systems is able to monitor all events taking place in their X-ray machines.

The three trends mentioned above are important enablers for path mining and process mining. The abundance of recorded events in structured format is an important enabler for the analysis of run-time behavior. Moreover, the desire to be flexible and adaptable also triggers the need for monitoring. If processes are not enforced by some system, it is relevant to find out what is actually happening, e.g., how frequently do people deviate from the default procedure.

BACKGROUND

Path mining can be seen as a tool in the context of Business Process Intelligence (BPI). This approach to path mining uses generic mining tools to extract implicit rules that govern the path of tasks followed during the execution of a process. Generally, the realization of a process can be carried out by executing a subset of tasks. Path mining is fundamentally about identifying the subset of tasks that will be potentially be triggered during the realization of a process. Path mining is important to process Quality of Service (QoS) prediction algorithms (Cardoso, Miller et al. 2004). In processes for e-commerce, suppliers and customers define a contract between the two parties, specifying QoS items such as products or services to be delivered, deadlines, quality of products, and cost of services. A process, which typically has a graph-like representation, includes a number of linearly independent control paths (i.e. paths that are

executed in parallel). Depending on the path followed during the execution of a process, the QoS may substantially be different. If we can predict with a certain degree of confidence the path that will be followed at runtime, we can significantly increase the precision of QoS estimation algorithms for processes.

Process mining has emerged as a way to analyze systems and their actual use based on the event logs they produce (Aalst, Dongen, et al. 2003; Aalst, Weijters, Maruster, 2004; Aalst, Reijers, et al. 2007). Note that, unlike classical data mining, the focus of process mining is on concurrent processes and not on static or mainly sequential structures. Process mining techniques attempt to extract non-trivial and useful information from event logs. One element of process mining is control-flow discovery, i.e., automatically constructing a process model (e.g., a Petri net) describing the causal dependencies between activities. In many domains, processes are evolving and people, typically, have an oversimplified and incorrect view on the actual business processes. Therefore, it is interesting to compare reality (as recorded in the log) with models. Since process mining is so important for organizations, there was the need to develop a system which implements the most significant algorithms developed up to date. Therefore, the ProM framework has been developed as a completely plug-able environment. Different research groups spread out over the world have contributed to ProM. Currently there are more than 150 plug-ins available, thus supporting all aspects of process mining (Aalst, Reijers, et al. 2007).

SETTINGS

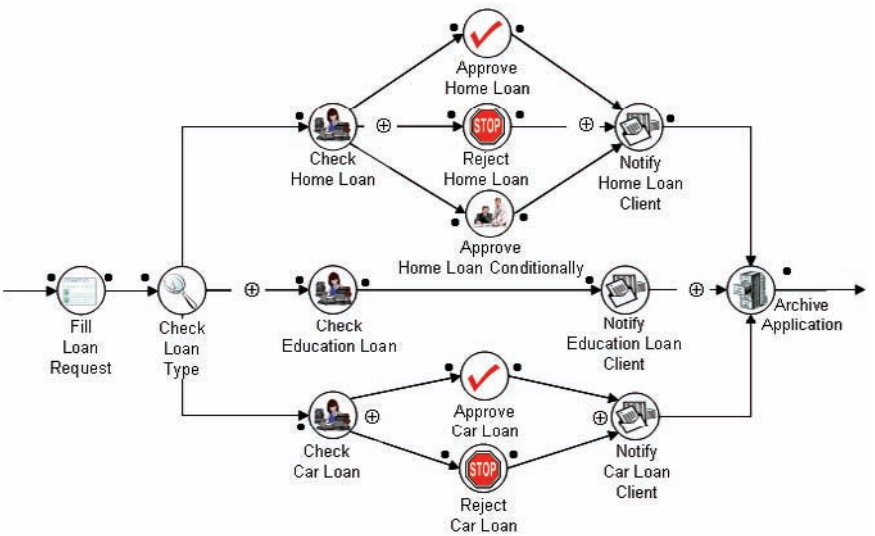
This section presents a typical business process model and illustrates also a typical process log. These two elements will be use to explain the concepts of path mining and process mining in the next section.

Business Process Scenario

A major bank has realized that to be competitive and efficient it must adopt a new and modern information system infrastructure. Therefore, a first step was taken in that direction with the adoption of a workflow management system to support its business processes. All the services available to customers are stored and executed under the supervision of the workflow system. One of the services supplied by the bank is the loan process depicted in Figure 1.

The process of the scenario is composed of fourteen tasks. For example, The Fill Loan Request task allows clients to request a loan from the bank. In this step, the client is asked to fill in an electronic form with personal information and data describing the condition of the loan being requested. The second task, Check Loan Type, determines the type of loan a client has requested

Figure 1. The loan process



and, based on the type, forwards the request to one of three tasks: Check Home Loan, Check Educational Loan, or Check Car Loan.

Process Log

Many systems have some kind of event log often referred to as "history", "audit trail", "transaction log", etc. (Agrawal, Gunopulos et al. 1998; Grigori, Casati et al. 2001; Sayal, Casati et al. 2002; Aalst, Dongen et al. 2003). The event log typically contains information about events referring to a task and a case. The case (also named process instance) is the "thing" which is being handled, e.g., a customer order, a job application, an insurance claim, or a building permit. Table 1 illustrates an example of a process log.

PATH MINING

To perform path mining, current process logs need to be extended to store information indicating the values and the type of the input parameters passed to tasks and the output parameters received from tasks. Table 2 shows an extended process log which accommodates input/output values of tasks parameters that have been generated at runtime. Each 'Parameter/Value' entry has a type, a parameter name, and a value (for example, string loan-type="car-loan").

Additionally, the process log needs to include path information, a path describing the tasks that have been executed during the enactment of a process. For example, in the process log illustrated in Table 2, the service NotifyUser is the last service of a process. The log has been extended in such a way that the NotifyUser record contains information about the path that has been followed during the process execution.

Process Profile

When beginning work on path mining, it is necessary to elaborate a profile for each process. A profile provides the input to machine learning and it is characterized by its values on a fixed, predefined set of attributes. The attributes correspond to the task input/output parameters that have been stored previously in the process log. Path mining will be performed on these attributes.

Profile Classification

The attributes present in a profile trigger the execution of a specific set of tasks. Therefore, for each profile previously constructed, we associate an additional attribute, the path attribute, indicating the path followed

Table 1. Process log

Date	Process	Case	Task	Task instance	Cost	Dur.	…
6:45 03-03-04	LoanApplication	LA04	RejectCarLoan	RCL03	$1.2	13 min	…
6:51 03-03-04	TravelRequest	TR08	FillRequestTravel	FRT03	$1.1	14 min	…
6:59 03-03-04	TravelRequest	TR09	NotifyUser	NU07	$1.4	24 hrs	…
7:01 03-03-04	InsuranceClaim	IC02	SubmitClaim	SC06	$1.2	05 min	…
…	…	…	…	…	…	…	…

Table 2. Extended process log

…	Case	Task	Task instance	Parameter/Value	Path	…
…	LA04	NotifyCLoanClient	NLC07	string e-mail="jf@uma.pt"	…	…
…	LA05	CheckLoanRequest	CLR05	double income=12000; string Name="Eibe Frank";	…	…
…	TR09	NotifyUser	NU07	String e-mail=jf@uma.pt; String telef="35129170023"	FillForm->CheckForm->Approve->Sign->Report	…
…	…	…	…		…	…

when the attributes of the profile have been assigned to specific values. The path attribute is a target class. Classification algorithms classify samples or instances into target classes. Once the profiles and a path attribute value for each profile have been determined, we can use data mining methods to establish a relationship between the profiles and the paths followed at runtime.

Experiments

In this section, we present the results of applying an algorithm to a synthetic loan dataset. To generate a synthetic dataset, we start with the process presented in the introductory scenario, and using this as a process model graph, log a set of process instance executions.

The data are lists of event records stored in a process log consisting of process names, instance identification, task names, variable names, etc.Table 3 shows the additional data that have been stored in the process log. The information includes the task variable values that are logged by the system and the path that has been followed during the execution of instances. Each entry corresponds to an instance execution.

Process profiles are characterized by a set of six attributes: income, loan_type, loan_amount, loan_years, name and SSN. These attributes correspond to the task input/output parameters that have been stored previously in the process log presented inTable 3. Each profile is associated with a class indicating the path that has been followed during the execution of a process when the attributes of the profile have been assigned specific values. The last column ofTable 3 shows the class named path.

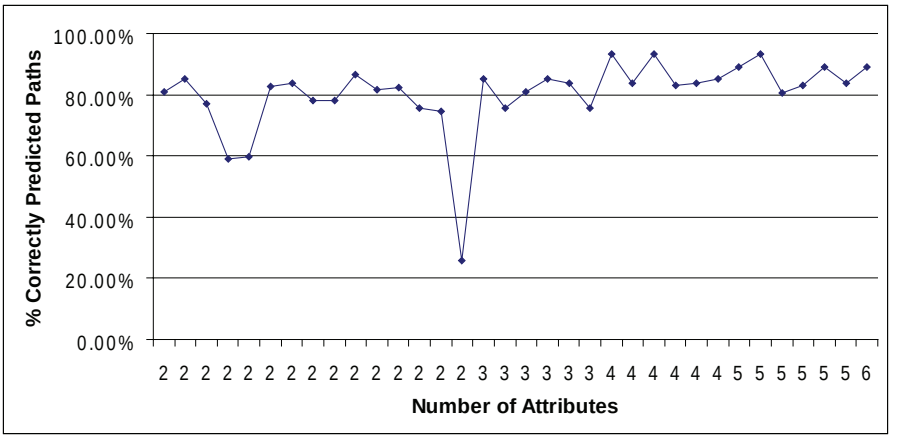
Once profiles are constructed and associated with paths, this data is combined and formatted to be analyzed using Weka (Weka 2004). We have used the J.48 algorithm, which is Weka's implementation of the C4.5 (Hand, Mannila et al. 2001; Weka 2004) decision tree learner to classify profiles.

Each experiment has involved data from 1000 process executions and a variable number of attributes (ranging from 2 attributes to 6 attributes). We have conducted 34 experiments, analyzing a total of 34000 records containing data from process instance executions. Figure 2 shows the results that we have obtained.

Table 3. Additional data stored in the process log

income	loan_type	Loan_ amount	loan_ years	name	SSN	Path
1361.0	Home-Loan	129982.0	33	Bernard-Boar	10015415	FR>CLT>CHL>AHL>NHC>CA
Unknown	Education-Loan	Unknown	Unknown	John-Miller	15572979	FR>CLT>CEL>CA
1475.0	Car-Loan	15002.0	9	Eibe-Frank	10169316	FR>CLT>CCL>ACL>NCC>CA
…	…	…	…	…	…	…

Figure 2. Experimental results



The path mining technique developed has achieved encouraging results. When three or more attributes are involved in the prediction, the system is able to predict correctly the path followed for more than 75% of the process instances. This accuracy improves when four attributes are involved in the prediction, in this case more than 82% of the paths are correctly predicted. When five attributes are involved, we obtain a level of prediction that reaches a high of 93.4%. Involving all the six attributes in the prediction gives excellent results: 88.9% of the paths are correctly predicted. When a small number of attributes are involved in the prediction, the results are not as good. For example, when only two attributes are selected, we obtain predictions that range from 25.9% to 86.7%.

PROCESS MINING

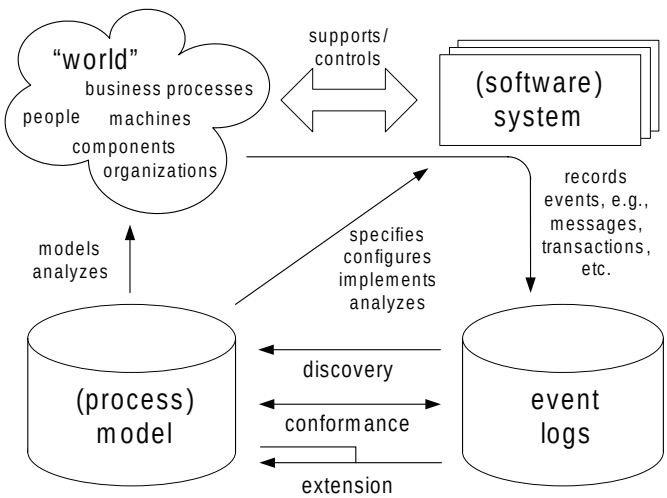
Assuming that we are able to log events, a wide range of process mining techniques comes into reach (Aalst, Dongen, et al. 2003; Aalst, Weijters, Maruster, 2004; Aalst, Reijers, et al. 2007). The basic idea of process mining is to learn from observed executions of a process and (1) to discover new models (e.g., constructing a Petri net that is able to reproduce the observed behavior), (2) to check the conformance of a model by checking whether the modeled behavior matches the observed behavior, and (3) to extend an existing model by projecting information extracted from the logs onto some initial model (e.g., show bottlenecks in a process model by analyzing the event log).

Process discovery: Traditionally, process mining has been focusing on discovery, i.e., deriving information about the original process model, the organizational context, and execution properties from enactment logs. An example of a technique addressing the control flow perspective is the alpha algorithm, which constructs a Petri net model (Aalst, Weijters, Maruster, 2004) describing the behavior observed in the event log. For example, based on a log like the one depicted in Table 1 it is possible to discover the process model shown in Figure 1. In fact to discover this process model, only the columns “Case” and “Task” are needed.

Conformance checking: Conformance checking compares an a-priori model with the observed behavior as recorded in the log. In (Rozinat, Aalst, 2005) it is shown how a process model (e.g., a Petri net) can be evaluated in the context of a log using metrics such as “fitness” (is the observed behavior possible according to the model?) and “appropriateness” (Is the model “typical” for the observed behavior?). In the example, we could compare the observed behavior in Table 1 with the modeled behavior in Figure 1.

Process extension: There are different ways to extend a given process model with additional perspectives

Figure 3. The three types of process mining (discovery, conformance, and extension) and their relations with models and event logs



based on event logs, e.g., decision mining (Rozinat, Aalst, 2006). Starting from a process model, one can analyze how data attributes influence the choices made in the process based on past process executions. A process model can be extended with timing information (e.g., bottleneck analysis). Clearly, decision mining is closely related to path mining. For example, it is possible to highlight bottlenecks in a discovered process model using the timestamps typically present in a log.

Figure 3 shows the three types of process mining introduced above (discovery, conformance, and extension). Each of the techniques involves a model and an event log. In case of discovery the process model is discovered on the basis of the data present in the log. In case of conformance and extension there is already some initial model.

ProM

In recent years ProM (www.processmining.org) has emerged as a broad and powerful process analysis tool, supporting all kinds of analysis related to business processes (such as verification and error checking). In contrast to many other analysis tools the starting point was the analysis of real processes rather than modeled processes, i.e., using *process mining* techniques ProM attempts to extract non-trivial and useful information from so-called "event logs".

Traditionally, most analysis tools focusing on processes are restricted to *model-based analysis*, i.e., a model is used as the starting point of analysis. For example, a purchasing process can be modeled using EPCs (Event-Driven Process Chains) and verification techniques can then be used to check the correctness of the protocol while simulation can be used to estimate performance aspects. Such analysis is *only useful if the model reflects reality*. Therefore, ProM can be used to both analyze models and logs. Basically, ProM supports all types of analysis mentioned before and the goal is to support the entire spectrum shown in Figure 3. It provides a plug-able environment for process mining offering a wide variety of plug-ins for process discovery, conformance checking, model extension, model transformation, etc. ProM is open source and can be downloaded from www.processmining.org. Many of its plug-ins work on Petri nets, e.g., there are several plug-ins to discover Petri nets using techniques ranging from genetic algorithms and heuristics to regions and partial orders. Moreover, Petri nets can be analyzed in various ways using the various analysis plug-ins. However, ProM allows for a wide variety of model types, conversions, and import and export facilities. For example, it is possible to convert Petri nets into BPEL, EPCs and YAWL models and vice versa.

FUTURE TRENDS

In the future, it is expected to see a wider spectrum of applications managing processes in organizations. According to the Aberdeen Group's estimates, spending in the Business Process Management software sector (which includes workflow systems) reached $2.26 billion in 2001 (Cowley 2002).

We are currently extending and improving mining techniques and continue to do so given the many challenges and open problems. For example, we are developing genetic algorithms to improve the mining of noisy logs. Recently, we added a lot of new functionality to the ProM framework. ProM is already able to mine from FLOWer logs and we applied this in a Dutch social security agency. Moreover, we have used ProM to analyze the processes of different hospitals, municipalities, governmental agencies, and banks. We use process mining to analyze the actual use of copiers, medical equipment, and wafer steppers. Furthermore, we are extending ProM with quality metrics to analyze business processes (Vanderfeesten, Cardoso et al. 2007; Mendling, Reijers et al. 2007).

CONCLUSION

Business Process Management Systems, processes, workflows, and workflow systems represent fundamental technological infrastructures that efficiently define, manage, and support business processes. The data generated from the execution and management of processes can be used to discover and extract knowledge about the process executions and structure.

We have shown that one important area of processes to analyze is path mining, i.e. the prediction of the path that will be followed during the execution of a process. From the experiments, we can conclude that classification methods are a good solution to perform path mining on administrative and production processes. We have also shown that business process mining aims at the extraction of knowledge from the behavior observed

in the event log. This type of mining can be used to, for example, construct a model based on an event log or to check if reality conforms to the model. Several process mining techniques have been implemented and are made available through the ProM framework.

REFERENCES

Aalst, W.M.P. van der, Dongen, B. F. v., Herbst, J., Maruster, J., Schimm, G. & Reijers, H.A. (2003). Workflow Mining: A Survey of Issues and Approaches. *Data & Knowledge Engineering*, 47(2), 237-267.

Aalst, W.M.P. van der, Reijers, H.A. & Song, M. (2005). Discovering Social Networks from Event Logs. *Computer Supported Cooperative Work*, 14(6), 549–593.

Aalst, W.M.P. van der, Reijers, H. A., Weijters, A.J.M.M., Dongen, B.F. van, Alves de Medeiros, A.K., Song, M. & Verbeek, H.M.W. (2007). Business process mining: An industrial application. *Information Systems*, 32(5), 713-732.

Aalst, W.M.P. van der, Weijters, A.J.M.M. & Maruster, L. (2004). Workflow Mining: Discovering Process Models from Event Logs. *IEEE Transactions on Knowledge and Data Engineering*, 16(9), 1128–1142.

Agrawal, R., Gunopulos, D. & Leymann, F. (1998). Mining Process Models from Workflow Logs. *Sixth International Conference on Extending Database Technology*, Valencia, Spain, Lecture Notes in Computer Science Vol. 1377, Springer, 469-483.

Cardoso, J., Miller, J., Sheth, A., Arnold, J. & Kochut, K. (2004). Modeling Quality of Service for workflows and web service processes. *Web Semantics: Science, Services and Agents on the World Wide Web Journal*, 1(3), 281-308.

Cowley, S. (2002). Study: BPM market primed for growth. Retrieved March 22, 2006, from *http://www.infoworld.com*

Dumas, M., Aalst, W.M.P. van der & A. H. t. Hofstede (2005). *Process Aware Information Systems: Bridging People and Software Through Process Technology*, New York: Wiley-Interscience.

Grigori, D., Casati, F., Dayal, U. & Shan, M. C. (2001, September). Improving Business Process Quality through Exception Understanding, Prediction, and Prevention. *27th VLDB Conference*, Roma, Italy, 2001.

Hand, D. J., Mannila, H. & Smyth, P. (2001). *Principles of Data Mining*. Bradford Book.

Mendling, J., Reijers, H. A. & Cardoso, J. (2007, September). What Makes Process Models Understandable? *Business Process Management 2007*. Brisbane, Australia, 2007, 48-63.

Sayal, M., Casati, F., Dayal, U. & Shan, M. C. (2002, August). Business Process Cockpit. *28th International Conference on Very Large Data Bases*, VLDB'02. Hong Kong, China, 880-883

.Smith, H. and Fingar, P. (2003). *Business Process Management (BPM): The Third Wave*. FL, USA, Meghan-Kiffer Press.

Vanderfeesten, I., Cardoso, J., Mendling, J., Reijers, H. & Aalst, W.M.P. van der. (2007). Quality Metrics for Business Process Models. In L. Fischer (ed.) *Workflow Handbook 2007*. FL, USA: Lighthouse Point, Future Strategies Inc.

Weka (2004). Weka. Retrieved May 12, 2005, from http://www.cs.waikato.ac.nz/ml/weka/.

KEY TERMS

Business Process: A set of one or more linked activities which collectively realize a business objective or goal, normally within the context of an organizational structure.

Business Process Management System: A Business Process Management System (BPMS) provides an organization with the ability to collectively define and model their business processes, deploy these processes as applications that are integrated with their existing software systems, and then provide managers with the visibility to monitor, analyze, control and improve the execution of those processes.

Process Definition: The representation of a business process in a form which supports automated manipulation or enactment by a workflow management system.

Process Log: During the execution of processes, events and messages generated by the enactment sys-

tem are recorded and stored in a process log. It is an electronic archive in which the history of instances is recorded. It contains various details about each instance, such as starting time, tasks performed and resources allocated.

Task: A task is an "atomic" process: one which is not further subdivided into component processes. It is thus a logical unit of work; in other words a task is either carried out in full or not at all.

Workflow: The automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules.

Workflow Engine: A workflow engine provides the actual management of workflows. Amongst other things, it is concerned with task-assignment generation, resource allocation, activity performance, the launching of applications and the recording of logistical information.

Workflow Management System: A system that defines, creates and manages the execution of workflows through the use of software which is able to interpret the process definition, interact with participants and, where required, invoke the use of tools and applications.

Pattern Discovery as Event Association

Andrew K. C. Wong
University of Waterloo, Canada

Yang Wang
Pattern Discovery Technology, Canada

Gary C. L. Li
University of Waterloo, Canada

INTRODUCTION

A basic task of machine learning and data mining is to automatically uncover **patterns** that reflect regularities in a data set. When dealing with a large database, especially when domain knowledge is not available or very weak, this can be a challenging task. The purpose of **pattern discovery** is to find non-random relations among events from data sets. For example, the "exclusive OR" (XOR) problem concerns 3 binary variables, A, B and C=A⊗B, i.e. C is true when either A or B, but not both, is true. Suppose not knowing that it is the XOR problem, we would like to check whether or not the occurrence of the compound event [A=T, B=T, C=F] is just a random happening. If we could estimate its frequency of occurrences under the random assumption, then we know that it is not random if the observed frequency deviates significantly from that assumption. We refer to such a compound event as an event association pattern, or simply a **pattern**, if its frequency of occurrences significantly deviates from the default random assumption in the statistical sense. For instance, suppose that an XOR database contains 1000 samples and each primary event (e.g. [A=T]) occurs 500 times. The expected frequency of occurrences of the compound event [A=T, B=T, C=F] under the independence assumption is $0.5\times0.5\times0.5\times1000 = 125$. Suppose that its observed frequency is 250, we would like to see whether or not the difference between the observed and expected frequencies (i.e. 250 – 125) is significant enough to indicate that the compound event is not a random happening.

In statistics, to test the correlation between random variables, **contingency table** with chi-squared statistic (Mills, 1955) is widely used. Instead of investigating variable correlations, pattern discovery shifts the traditional correlation analysis in statistics at the variable level to association analysis at the event level, offering an effective method to detect statistical association among events.

In the early 90's, this approach was established for second order event associations (Chan & Wong, 1990). A higher order **pattern discovery** algorithm was devised in the mid 90's for discrete-valued data sets (Wong & Yang, 1997). In our methods, patterns inherent in data are defined as statistically significant associations of two or more primary events of different attributes if they pass a statistical test for deviation significance based on **residual analysis**. The discovered high order patterns can then be used for classification (Wang & Wong, 2003). With continuous data, events are defined as Borel sets and the pattern discovery process is formulated as an optimization problem which recursively partitions the sample space for the best set of significant events (patterns) in the form of high dimension intervals from which probability density can be estimated by Gaussian kernel fit (Chau & Wong, 1999). Classification can then be achieved using Bayesian classifiers. For data with a mixture of discrete and continuous data (Wong & Yang, 2003), the latter is categorized based on a global optimization discretization algorithm (Liu, Wong & Yang, 2004). As demonstrated in numerous real-world and commercial applications (Yang, 2002), pattern discovery is an ideal tool to uncover subtle and useful patterns in a database.

In pattern discovery, three open problems are addressed. The first concerns learning where noise and uncertainty are present. In our method, noise is taken as inconsistent samples against statistically significant patterns. Missing attribute values are also considered as noise. Using a standard statistical **hypothesis testing** to confirm statistical patterns from the candidates, this method is a less ad hoc approach to discover patterns than most of its contemporaries. The second problem

concerns the detection of polythetic patterns without relying on exhaustive search. Efficient systems for detecting monothetic patterns between two attributes exist (e.g. Chan & Wong, 1990). However, for detecting polythetic patterns, an exhaustive search is required (Han, 2001). In many problem domains, polythetic assessments of feature combinations (or higher order relationship detection) are imperative for robust learning. Our method resolves this problem by directly constructing polythetic concepts while screening out non-informative pattern candidates, using statistics-based heuristics in the discovery process. The third problem concerns the representation of the detected patterns. Traditionally, if-then rules and graphs, including networks and trees, are the most popular ones. However, they have shortcomings when dealing with multilevel and multiple order patterns due to the non-exhaustive and unpredictable hierarchical nature of the inherent patterns. We adopt **attributed hypergraph** (AHG) (Wang & Wong, 1996) as the representation of the detected patterns. It is a data structure general enough to encode information at many levels of abstraction, yet simple enough to quantify the information content of its organized structure. It is able to encode both the qualitative and the quantitative characteristics and relations inherent in the data set.

BACKGROUND

In the ordinary sense, "discovering regularities" from a system or a data set implies partitioning the observed instances into classes based on similarity. Michalski and Stepp (1983) pointed out that the traditional distance-based statistical clustering techniques make no distinction among relevant, less relevant and irrelevant attributes nor do they render conceptual description of the clusters with human input. They proposed CLUSTER/2 as a conceptual clustering algorithm in a noise-free environment. It is effective for small data sets containing no noise yet computationally expensive for a large data set even with its Hierarchy-building Module. To deal with noise, COBWEB was introduced by Fisher (1987). However, the concept tree generated by COBWEB might be very large. For deterministic pattern discovery problems such as the MONK, COBWEB does not work well when compared with other AI and connectionist approaches (Han, 2001).

The Bayesian methods provide a framework for reasoning with partial beliefs under uncertainty. To perform inferences, they need to estimate large matrices of probabilities for the network during training (Pearl, 1988). When going to high-order cases, the contingency table introduces a heavy computation load.

Agrawal and Srikant (1994) proposed association rule mining to detect relationship among items in transactional database. It is well-suited to applications such as market basket analysis but not applicable in some other applications such as capturing correlations between items where association rules may be misleading (Han, 2001). Hence, Brin, Motwani and Silverstein (1997) proposed to detect correlation rules from the contingency table. However, correlation rule mining is not accurate when the contingency table is sparse or larger than 2×2 (Han, 2001) since it is designed for testing the correlation of two random variables.

Pattern discovery shifts the statistical test from the entire **contingency table** to the individual cells in the table. A **hypothesis test** at each individual cell is formalized by **residual analysis**. Therefore, it handles sparse and high dimensional contingency table much more effectively.

Recent development in pattern discovery methodologies includes building classifiers based on the discovered patterns. A typical example is Liu, Hsu and Ma 's CBA (1998) that uses association rules to classify data. More recent works include HWPR (Wang & Wong, 2003) and DeEPs (Li et. al., 2004). HWPR employs event associations as classification rules whereas DeEPs uses emerging patterns, a variation of association rules, to classify data. A detailed evaluation and comparison of these methods can be found in (Sun et. al., 2006).

MAIN FOCUS

Event Associations

Consider a data set D containing M data samples. Every sample is described by N attributes, each of which can assume values from its own finite discrete alphabet. Let $\mathbf{X} = \{ X_1, \ldots, X_N \}$ represent this attribute set. Then, each attribute, X_i, $1 \leq i \leq N$, can be seen as a random variable taking on values from its alphabet $\alpha_i = \{\alpha_i^1, ..., \alpha_i^{m_i}\}$, where m_i is the cardinality of the alphabet of the ith attribute. Thus, a realization of $\mathbf{X}$ can be denoted by $\mathbf{x} = \{ x_1, \ldots, x_N \}$, where x_i can assume

any value in α_i. In this manner, each data sample **x** is a realization of **X**.

Definition 1: *A primary event of a random variable X_i ($1 \le i \le N$) is a realization of X_i taking on a value from α_i.*

The pth ($1 \le p \le m_i$) primary event of X_i can be denoted as

$$[X_i = \alpha_i^p]$$

or simply x_{ip}. We use x_i to denote the realization of X_i.

Let **s** be a subset of integers $\{1, \ldots, N\}$ containing k elements ($k \le N$) and $\mathbf{X}^s$ be a subset of **X** such that

$$\mathbf{X}^s = \{X_i \mid i \in \mathbf{s}\}$$

Then $\mathbf{x}^s_p$ represents the pth realization of $\mathbf{X}^s$. We use $\mathbf{x}^s$ to denote a realization of $\mathbf{X}^s$.

Definition 2: *A compound event associated with the variable subset $\mathbf{X}^s$ is a set of primary events instantiated by a realization $\mathbf{x}^s$ with order $|\mathbf{s}|$.*

A 1-compound event is a primary event. A k-compound event is made up of k primary events of k variables. Every sample is an N-compound event. For example, in the XOR problem, [A=T] is a primary event, [A=F, C=T] is a two-compound event, and, [A=T, B=F, C=T] is a three-compound event.

Definition 3: *Let T be a statistical significance test. If the frequency of occurrences of a compound event $\mathbf{x}^s$ is significantly deviated from its expectation based on a default probabilistic model, we say that $\mathbf{x}^s$ is a significant association pattern, or a pattern of order $|\mathbf{s}|$.*

The default model in definition 3 is the hypothesis that the variables in $\mathbf{X}^s$ are mutually independent. We then use the *standardized residual* (Haberman, 1974) to test the significance of its occurrence against the independence assumption. In the XOR example in the introduction section, the standardized residual of the compound event [A=T, B=T, C=F] is 11.18, larger than 1.96 which is the value at the 95% significant level. Thus, we conclude that the compound event *significantly deviates* from its independence expectation. Therefore, this compound event is a third-order pattern.

Let us denote the observed occurrences of compound event $\mathbf{x}^s$ as $o_{\mathbf{x}^s}$ and its expected occurrences as $e_{\mathbf{x}^s}$. Then,

$$e_{\mathbf{x}^s} = M \prod_{i \in \mathbf{s}, x_i \in \mathbf{x}^s} P(x_i) \qquad (1)$$

where $P(x_i)$ is estimated by the proportion of the occurrence of x_i to the sample size M.

To test whether or not $\mathbf{x}^s$ is a significant association pattern, *standardized residual* defined in (Haberman, 1974) is used to measure the deviation between $o_{\mathbf{x}^s}$ and $e_{\mathbf{x}^s}$:

$$z_{\mathbf{x}^s} = \frac{o_{\mathbf{x}^s} - e_{\mathbf{x}^s}}{\sqrt{e_{\mathbf{x}^s}}} \qquad (2)$$

Standardized residual $z_{\mathbf{x}^s}$ is the square root of χ^2. It has an asymptotic normal distribution with a mean of approximately zero and a variance of approximately one. Hence, if the value of $z_{\mathbf{x}^s}$ exceeds 1.96, by conventional criteria, we conclude that the primary events of $\mathbf{x}^s$ are "associated" and likely to occur together, with a confidence level of 95 percent. In this case, $\mathbf{x}^s$ is referred to as a positive pattern. If $z_{\mathbf{x}^s}$ is less than -1.96, we conclude that the primary events are unlikely to co-occur. It is referred to as a negative pattern.

Standardized residual is considered to be of normal distribution only when the asymptotic variance of $z_{\mathbf{x}^s}$ is close to one, otherwise, it has to be adjusted by its variance for a more precise analysis. The adjusted residual is expressed as:

$$d_{\mathbf{x}^s} = \frac{z_{\mathbf{x}^s}}{\sqrt{v_{\mathbf{x}^s}}} \qquad (3)$$

where $v_{\mathbf{x}^s}$ is the maximum likelihood estimate of the variance of $z_{\mathbf{x}^s}$ (Wong & Wang, 1997). To avoid exhaustive search, two criteria for eliminating impossible pattern candidates are used. One is the validation of a significance test and the other is testing of higher order negative patterns. An efficient "test-and-discard" search algorithm was developed to prune the searching space (see Wong and Wang (1997) for its complexity analysis).

As an illustration, we apply **pattern discovery** to an XOR data set containing 102 samples with 15% noise randomly added. Figure 1 shows all the patterns detected. Here, the four compound events are found to be

Figure 1. Patterns discovered from the XOR data set with noises

Index	Residual	Probability	Order	A	B	C	
0	3.973630	0.23	3	F	F	F	
1	3.279978	0.19	3	T	F	T	
2	2.784015	0.2	3	F	T	T	+ve
3	1.990939	0.18	3	T	T	F	
4	-2.014950	0.07	3	T	T	T	
5	-2.746582	0.07	3	F	T	F	
6	-3.228580	0.04	3	T	F	F	-ve
7	-4.037667	0.02	3	F	F	T	

statistically significant from their expectations positively, and four others negatively. The output reflects exactly the XOR relationship, e.g. the positive pattern [A=F, B=F, C=F] (the first pattern) states that C=A⊗B=F if A=F and B=F. Furthermore, the negative pattern [A=T, B=T, C=T] (the fifth pattern) states that three events will not occur at the same time. More details about the XOR experiments and others by pattern discovery can be found in Wong and Wang (1997).

Classification based on Event Associations

A well-developed **classification** method is built using the discovered patterns (Wang & Wong, 2003). It generates classification rules by using *weight of evidence* to quantify the evidence of significant association patterns in support of, or against a certain class membership.

In information theory, the difference in the gain of information when predicting attribute Y taking on the value y_i against taking on other values, given $\mathbf{x}$, is a measure of evidence provided by $\mathbf{x}$ in favor of y_i being a plausible value of Y as opposed to Y taking other values. This difference, denoted by $W(Y = y_i / Y \neq y_i \mid \mathbf{x})$, is defined as the weight of evidence:

$$W(Y = y_i / Y \neq y_i \mid \mathbf{x}) = I(Y = y_i : \mathbf{x}) - I(Y \neq y_i : \mathbf{x})$$

$$= \log \frac{P(Y = y_i \mid \underline{\mathbf{x}})}{P(Y = y_i)} - \log \frac{P(Y \neq y_i \mid \underline{\mathbf{x}})}{P(Y \neq y_i)}$$

$$= \log \frac{P(\underline{\mathbf{x}} \mid Y = y_i)}{P(\underline{\mathbf{x}} \mid Y \neq y_i)} \qquad (4)$$

where $I(\cdot)$ is the mutual information. It is positive if $\mathbf{x}$ provides positive evidence supporting Y taking on y_i, otherwise, it is negative, or zero.

Suppose that n sub-compound events $\mathbf{x}_1,\ldots,\mathbf{x}_n$ are detected, where $(\mathbf{x}_k, Y = y_i)$ is a significant event association and

$$\bigcup_{k=1}^{n} \underline{\mathbf{x}}_k = \underline{\mathbf{x}};$$

$$\underline{\mathbf{x}}_p \bigcap \underline{\mathbf{x}}_q = \varphi$$

when $p \neq q$, $1 \leq k, p, q \leq n$. According to Wang and Wong (2003), $W(Y = y_i / Y \neq y_i \mid \mathbf{x})$ can be obtained from the sum of the weights of evidence for all event associations in $\mathbf{x}$ supporting $Y = y_i$. That is

$$W(Y = y_i / Y \neq y_i \mid \mathbf{x}) =$$

$$\log \frac{P(\underline{\mathbf{x}}_1 \mid Y = y_i)}{P(\underline{\mathbf{x}}_1 \mid Y \neq y_i)} + \ldots + \log \frac{P(\underline{\mathbf{x}}_n \mid Y = y_i)}{P(\underline{\mathbf{x}}_n \mid Y \neq y_i)}$$

$$= W(Y = y_i / Y \neq y_i \mid \mathbf{x}_1) + \ldots + W(Y = y_i / Y \neq y_i \mid \mathbf{x}_n)$$

$$= \sum_{k=1}^{n} W(Y = y_i \ / \ Y \neq y_i \mid \underline{\mathbf{x}}_k) \qquad (5)$$

Thus, the calculation of weight of evidence is to find a proper set of disjoint significant event associations from $\mathbf{x}$ and to sum individual weights of evidence provided by each of them. The task of **classification** is to maximize the term in Equation (5) so as to find the most plausible value y_i of Y with the highest weight.

As an illustration and a comparison with other methods, we apply our classification method to the mushroom database from UCI (Murph & Aha, 1987) with 8,124 samples characterized by 23 attributes, including the class label (edible, poisonous). We conducted the experiment in two ways. First, a subset of 500 instances was randomly sampled from the original 8,124 instances. Of the 500, 400 (80%) are used in the discovery phase while the remaining 100 (20%) are used for testing. This process is repeated for 10 times to obtain an average performance. In the 10 trials, the lowest classification accuracy we obtain is 97% and the highest is 100%. The average accuracy is 98.9%. In the second setting, the original 8,124 instances are divided into a subset of 5,416 (66.7%) for pattern discovery and a subset of 2,708 (33.3 %) for classification testing. The accuracy for this one-trial classification is 99.1%. These performances are consistent with others using the same data set. Table 1 is a comparison with the others' results available with the data set in UCI repository. We also ran two common associative classifiers CBA (Liu, Hsu and Ma, 1998) and DeEPs (Li et. al., 2004) and a conventional decision tree C4.5 classifier on the mushroom data. More detailed comparisons can be found in (Sun et. al., 2006).

A specific characteristic of our method is that the rules used for classification can be made explicit for interpretation and further analysis. On the one hand, if we select a particular rule, its detailed usage for classification can be displayed. For example, figure 2 shows that rule 25032 correctly classifies samples 263, 264, etc; and incorrectly classifies samples 300, 326, etc. On the other hand, if we select a particular sample, its detailed classification information can be shown. Figure 3 shows that sample 8 is supported by 8 rules (rules 1108, 25032, etc) as belonging to class edible with infinity weight of evidence.

FUTURE TRENDS

The ultimate goal of automatic knowledge acquisition is to discover useful knowledge inherent in data sets automatically. Once patterns of different orders are discovered, to extract knowledge from patterns, they should be organized and represented in a form appropriate for further analysis and interpretation. Ongoing research in pattern discovery focuses on methods for effective organization and representation of discovered patterns, especially in situations when the number of discovered patterns is enormous. A method known as **pattern clustering** is developed which simultaneously clusters high order patterns and their associated data (Wong & Li, in preparation). It uses an information theoretic measure to reflect the change of uncertainty before and after merging two pattern-associated data sets. At the end of the process, patterns are grouped together into pattern clusters; and their associated data are grouped according to the patterns. The relations among data clusters formed via pattern clustering are then represented in an **attributed hypergraph** which is lucid for pattern interpretation and analysis (Wang & Wong 1996). From the organized relations among data clusters, flexible visualization techniques are under development to enable users to

Table 1. Classification results for mushroom data

Classifier	Training Sample#	Classification Accuracy	Note
STAGGER	1000	95%	asymptoted
HILLARY	1000	95%	ten-trial average
CBA	5416	97.8%	one-trial
DeEPs	5416	99.0%	one-trial
C4.5	5416	99.3%	one-trial
Our method	500	98.9%	ten-trial average
Our method	5416	99.1%	one-trial

Figure 2. The detailed classification usage of rule 235

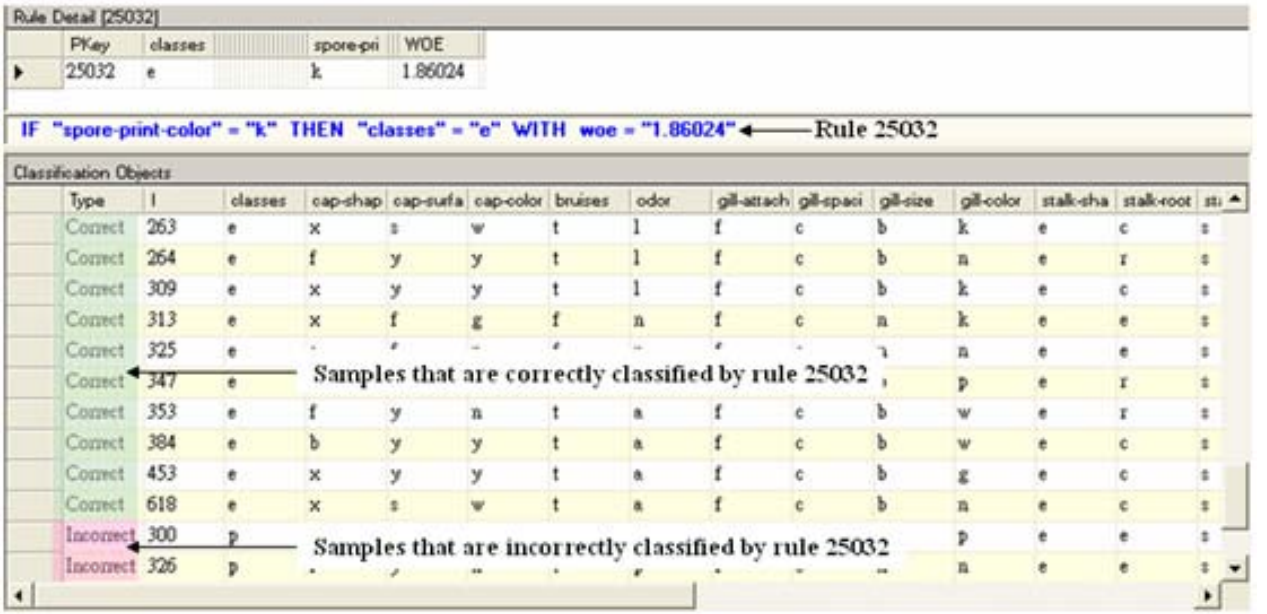
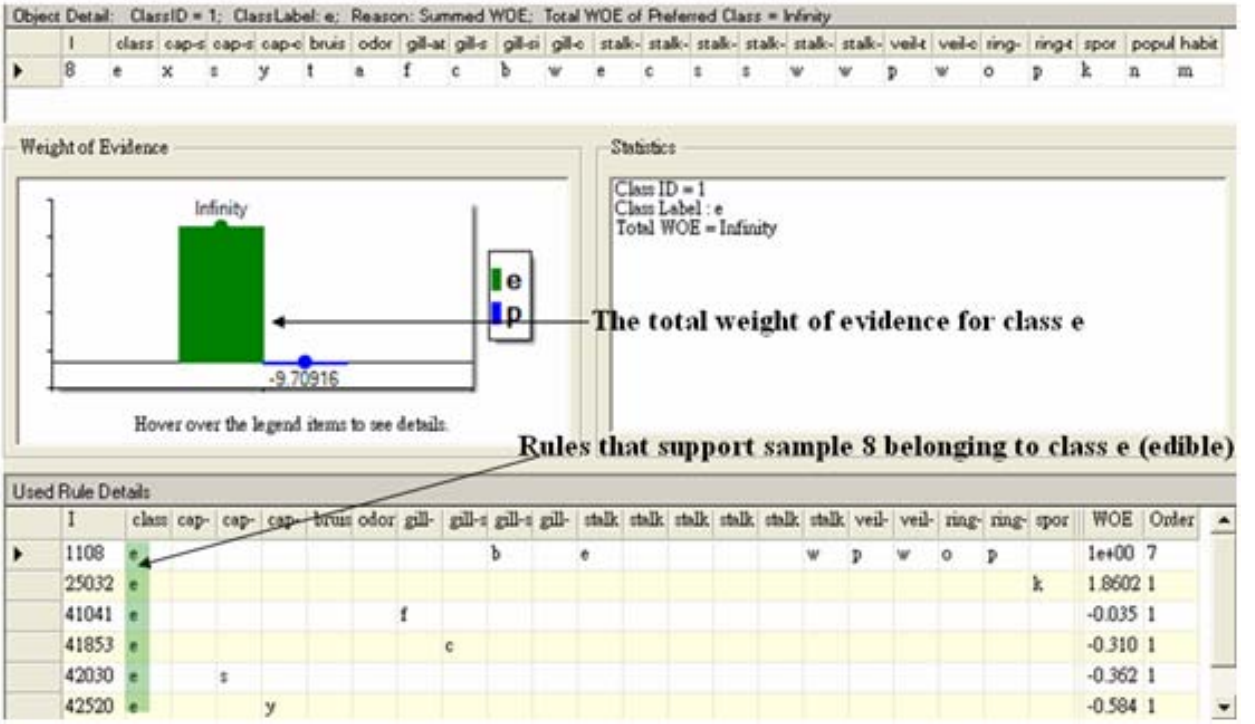


Figure 3. The detailed classification information of sample 8



acquire and display the discovered patterns in an effective ways.

In the applications aspect, with its ability to handle mixed mode data, it is generic to handle a great variety of problems. Because it can automatically discover association patterns of residues, genes and gene expression levels, it will be an ideal tool to uncover probabilistic patterns inherent in genomic and microarray data. It is able to relate data groups with similar associated patterns

to drug discovery as well as diagnostic, therapeutic and prognostic supports in healthcare.

CONCLUSION

The fundamental questions pattern discovery methodologies addresses are "what is a pattern?" and "where are the patterns?" By defining patterns as statistical significant event associations, pattern discovery renders an effective and efficient method to automatically discover patterns wherever they are in a data set. The discovered patterns can be used in decision support such as classification, clustering and density estimation. When the discovered patterns and their relations are made explicit, users would have a better understanding of how events are associated and related. Thus pattern discovery provides a means to its users for in-depth understanding, comprehensive interpretation and automatic yet systematic organization of statistically significant associated events with empirical supports that help the growth and reliable use of knowledge.

REFERENCES

Agrawal, R., & Srikant, R. (1994). Fast Algorithms for Mining Association Rules. *In Proceedings of the 20th International Conference on Very Large Data Bases* (pp. 487–499). Santiago, Chile.

Brin, S., Motwani, R., & Silverstein, R. (1997). Beyond Market Basket: Generalizing Association Rules to Correlations. *In Proceedings of ACM SIGMOD Conference on Management of Data* (pp. 265-276). Tucson, AZ.

Chan, K. C. C., & Wong, A. K. C. (1990). APACS: A systems for automated pattern analysis and classification. *Computation Intelligence, 6(3),* 119–131.

Chau, T., & Wong, A. K. C. (1999). Pattern Discovery by Residual Analysis and Recursive Partitioning. *IEEE Transactions on Knowledge & Data Engineering, 11(6),* 833-852.

Fisher, D. H. (1987). Knowledge Acquisition via Incremental Conceptual Clustering, *Machine Learning, 2(2),* 139-172.

Haberman, S. J. (1974). *The Analysis of Frequency Data*, University of Chicago Press.

Han, J. (2001). *Data Mining: Concepts and Techniques*, Morgan Kaufmann.

Li, J., Dong, G., Ramamohanarao, K., & Wong, L. (2004). DeEPs: a new instance-based lazy discovery and classification system. *Machine Learning, 54(2),* 99-124.

Liu, B., Hsu, W., & Ma, Y. (1998). Integrating classification and association rule mining. *In Proceedings of the Fourth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 80-86). New York.

Liu, L., Wong, A. K. C., & Wang, Y. (2004). A Global Optimal Algorithm for Class-Dependent Discretization of Continuous Data. *Intelligent Data Analysis, 8(2),* 151-170.

Michalski, R., & Stepp, P. (1983). Automated Construction of Classifications: Conceptual Clustering versus Numerical Taxonomy. *IEEE Transactions Pattern Analysis and Machine Intelligence, 5(4),* 396-409.

Mills, F. (1955). *Statistical Methods*, Pitman

Murph P. M., & Aha D. W. (1987). *UCI Repository of Machine Learning Databases*. Department of Information and Computer Science, University of California, Irvine.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann.

Sun, Y., Wong, A. K. C., & Wang Y. (2006). An Overview of Associative Classifier. *In Proceeding of International Conference on Data Mining* (pp. 88-87). Las Vegas, Nevada.

Wang, Y. (2002). Data Mining and Discover*E. *White paper on Pattern Discovery Software Systems Ltd. Data Mining Technology*, from http://www.patterndiscovery.com/eng/tech/discovere_whitepaper.pdf.

Wang, Y & Wong, A. K. C. (1996). Representing Discovered Patterns Using Attributed Hyperraph. *In Proceedings of International Conference on Knowledge Discovery and Data Mining* (pp. 283-286). Portland.

Wang, Y., & Wong, A. K. C. (2003). From Association to Classification: Inference Using Weight of Evidence. *IEEE Transactions on Knowledge & Data Engineering, 15(3)*, 914-925.

Wong, A. K. C., & Li, G. C. L. (in preparation). Pattern Clustering, Data Grouping and Data Characterization.

Wong, A. K. C., & Wang, Y (1997). High Order Pattern Discovery from Discrete-Valued Data. *IEEE Transactions on Knowledge & Data Engineering, 9(6)*, 877-893.

Wong, A. K. C., & Wang, Y. (2003). Pattern Discovery: A Data Driven Approach to Decision Support. *IEEE Transactions on System, Man, Cybernetic – Part C, 33(1)*, 114-124.

KEY TERMS

Associative Classification: A classification method based on association patterns/rules.

Attributed Hypergraph: A hypergraph whose vertices and hyperedges have attributes.

Event Association Pattern: A compound event whose occurrence significantly deviates from the random model.

Negative Pattern: An event association pattern with negative residual.

Pattern Clustering: An automatic dual process that simultaneously clusters event association patterns and their associated data.

Pattern Discovery: An methodology for automatically discovering non-random relations inherent among events from data sets.

Positive Pattern: An event association pattern with positive residual.

Pattern Preserving Clustering

Hui Xiong
Rutgers University, USA

Michael Steinbach
University of Minnesota, USA

Pang-Ning Tan
Michigan State University, USA

Vipin Kumar
University of Minnesota, USA

Wenjun Zhou
Rutgers University, USA

INTRODUCTION

Clustering and association analysis are important techniques for analyzing data. Cluster analysis (Jain & Dubes, 1988) provides insight into the data by dividing objects into groups (clusters), such that objects in a cluster are more similar to each other than to objects in other clusters. Association analysis (Agrawal, Imielinski & Swami, 1993), on the other hand, provides insight into the data by finding a large number of strong patterns -- frequent itemsets and other patterns derived from them -- in the data set. Indeed, both clustering and association analysis are concerned with finding groups of *strongly related* objects, although at different levels. Association analysis finds strongly related objects on a local level, i.e., with respect to a subset of attributes, while cluster analysis finds strongly related objects on a global level, i.e., by using all of the attributes to compute similarity values.

Recently, Xiong, Tan & Kumar (2003) have defined a new pattern for association analysis -- the hyperclique pattern -- which demonstrates a particularly strong connection between the overall similarity of all objects and the itemsets (local pattern) in which they are involved. The hyperclique pattern possesses a *high affinity property*: the objects in a hyperclique pattern have a guaranteed level of global pairwise similarity to one another as measured by the cosine similarity (uncentered Pearson correlation coefficient). Since clustering depends on similarity, it seems reasonable that the hyperclique pattern should have some connection to clustering.

Ironically, we found that hyperclique patterns are mostly destroyed by standard clustering techniques, i.e., standard clustering schemes do not preserve the hyperclique patterns, but rather, the objects comprising them are typically split among different clusters. To understand why this is not desirable, consider a set of hyperclique patterns for documents. The high affinity property of hyperclique patterns requires that these documents must be similar to one another; the stronger the hyperclique, the more similar the documents. Thus, for strong patterns, it would seem desirable (from a clustering viewpoint) that documents in the same pattern end up in the same cluster in many or most cases. As mentioned, however, this is not what happens for traditional clustering algorithms. This is not surprising since traditional clustering algorithms have no built in knowledge of these patterns and may often have goals that are in conflict with preserving patterns, e.g., minimize the distances of points from their closest cluster centroids.

More generally, the breaking of these patterns is also undesirable from an application point of view. Specifically, in many application domains, there are fundamental patterns that dominate the description and analysis of data within that area, e.g., in text mining, collections of words that form a topic, and in biological sciences, a set of proteins that form a functional module (Xiong et al. 2005). If these patterns are not

respected, then the value of a data analysis is greatly diminished for end users. If our interest is in patterns, such as hyperclique patterns, then we need a clustering approach that preserves these patterns, i.e., puts the objects of these patterns in the same cluster. Otherwise, the resulting clusters will be harder to understand since they must be interpreted solely in terms of objects instead of well-understood patterns.

There are two important considerations for developing a pattern persevering clustering approach. First, in any clustering scheme, we must take as our starting point the sets of objects that comprise the patterns of interest. If the objects of a pattern of interest are not together when we start the clustering process, they will often not be put together during clustering, since this is not the goal of the clustering algorithm. Second, if we start with the sets of objects that comprise the patterns of interest, we must not do anything in the clustering process to breakup these sets. Therefore, for pattern preserving clustering, the pattern must become the basic *object* of the clustering process. In theory, we could then use any clustering technique, although modifications would be needed to use sets of objects instead of objects as the basic starting point. Here, we use hyperclique patterns as our patterns of interest, since they have some advantages with respect to frequent itemsets for use in pattern preserving clustering: hypercliques have the high affinity property, which guarantees that keeping objects together makes sense, and finding hypercliques is computationally much easier than finding frequent itemsets. Finally, hyperclique patterns, if preserved, can help cluster interpretation.

BACKGROUND

Cluster analysis has been the focus of considerable work, both within data mining and in other fields such as statistics, psychology, and pattern recognition. Several recent surveys may be found in Berkhin (2002), Han, Kamber & Tung (2001), and Jain, Murty & Flynn (1999), while more extensive discussions of clustering are provided by the following books (Anderberg, 1973; Jain & Dubes, 1988; Kaufman & Rousseeuw, 1990).

While there are innumerable clustering algorithms, almost all of them can be classified as being either *partitional*, i.e., producing an un-nested set of clusters that partitions the objects in a data set into disjoint groups, or *hierarchical*, i.e., producing a nested sequence of partitions, with a single, all-inclusive cluster at the top and singleton clusters of individual points at the bottom. While this standard description of hierarchical versus partitional clustering assumes that each object belongs to a single cluster (a single cluster within one level, for hierarchical clustering), this requirement can be relaxed to allow clusters to overlap.

Perhaps the best known and widely used partitional clustering technique is K-means (MacQueen, 1999), which aims to cluster a dataset into K clusters---K specified by the user---so as to minimize the sum of the squared distances of points from their closest cluster centroid. (A cluster centroid is the mean of the points in the cluster.) K-means is simple and computationally efficient, and a modification of it, bisecting K-means (Steinbach, Karypis & Kumar, 2000), can also be used for hierarchical clustering.

Traditional hierarchical clustering approaches (Jain & Dubes, 1988) build a hierarchical clustering in an *agglomerative* manner by starting with individual points or objects as clusters, and then successively combining the two most similar clusters, where the similarity of two clusters can be defined in different ways and is what distinguishes one agglomerative hierarchical technique from another. These techniques have been used with good success for clustering documents and other types of data. In particular, the agglomerative clustering technique known as Group Average or UPGMA, which defines cluster similarity in terms of the average pairwise similarity between the objects in the two clusters, is widely used because it is more robust than many other agglomerative clustering approaches.

As far as we know, there are no other clustering methods based on the idea of preserving patterns. However, we mention two other types of clustering approaches that share some similarity with what we are doing here: constrained clustering and frequent item set based clustering. Constrained clustering (Wagstaff and Cardie, 2000, Tung, Ng, Lakshmanan & Han, 2001, Davidson and Ravi, 2005) is based on the idea of using standard clustering approaches, but restricting the clustering process. Our approach can be viewed as constraining certain objects to stay together during the clustering process. However, our constraints are automatically enforced by putting objects in hypercliques together, before the clustering process begins, and thus, the general framework for constrained clustering is not necessary for pattern preserving clustering.

Pattern preserving clustering techniques are based on an association pattern, the hyperclique pattern, but there have been other clustering approaches that have used frequent itemsets or other patterns derived from them. One of these approaches is Frequent Itemset-based Hierarchical Clustering (FIHC) (Benjamin, Fung & Wang 2003), which starts with clusters built around frequent itemsets. However, while our approach to clustering objects finds hypercliques of objects, and then uses them as initial clusters, FIHC uses selected itemsets -- sets of binary attributes -- to group objects (transactions), i.e., any object that supports an itemset goes into the same cluster. Thus, the approach of FIHC is quite different from pattern preserving clustering techniques, as is the approach of other clustering algorithms that are also based on frequent itemsets, e.g., Beil, Ester & Xu, 2002; Wang, Xu & Liu, 1999.

MAIN FOCUS

As the inspiration for pattern preserving clustering, the hyperclique pattern has been used to explore pattern preserving clustering. The concept of hyperclique patterns (Xiong, Tan & Kumar, 2003, Xiong, Tan & Kumar 2006) is based on the concepts of the frequent itemset and association rule (Agrawal, Imielinski & Swami, 1993).

Hyperclique Patterns

Let $I = \{i_1, i_2, ..., i_m\}$ be a set of distinct items. Each transaction T in database D is a subset of I. We call $X \subseteq I$ an itemset. The support of X, denoted by $supp(X)$, is the fraction of transactions containing X. If $supp(X)$ is no less than a user-specified minimum support, X is called a frequent itemset. The confidence of the association rule $X_1 \rightarrow X_2$ is defined as $conf(X_1 \rightarrow X_2) = supp(X_1 \cup X_2)/supp(X_1)$. It is the fraction of the transactions containing the items in X_1 that also contain the items in X_2, and is an estimate of the conditional probability of X_2 given X_1.

A hyperclique pattern (Xiong, Tan & Kumar, 2003) is a new type of association pattern that contains items that are *highly affiliated* with each other. By high affiliation, we mean that the presence of an item in a transaction strongly implies the presence of every other item that belongs to the same hyperclique pattern. The h-confidence measure (Xiong, Tan & Kumar, 2003) is specifically designed to capture the strength of this association. Formally, the h-confidence of an itemset $P = \{i_1, i_2, ..., i_m\}$ is defined as $hconf(P) = min_{1 \leq k \leq m}\{conf(i_k \rightarrow P - i_k)\}$. Given a set of items I and a minimum h-confidence threshold h_c, an itemset $P \subseteq I$ is a hyperclique pattern if and only if $hconf(P) \geq h_c$. For a hyperclique pattern P, the presence of any item $i \in P$ in a transaction implies the presence of all other items $P - \{i\}$ in the same transaction with probability at least h_c. This suggests that h-confidence is useful for capturing patterns containing items which are strongly related to each other. A hyperclique pattern is a maximal hyperclique pattern if no superset of this pattern is a hyperclique pattern.

Table 1 shows some hyperclique patterns identified from words of the LA1 dataset, which is part of the TREC-5 collection (http://trec.nist.gov) and includes articles from various news categories such as 'financial,' 'foreign,' 'metro,' 'sports,' and 'entertainment.' One hyperclique pattern listed in the table is {*mikhail*, *gorbachev*}, who is the former president of the Soviet

Table 1. Examples of hyperclique patterns from words of the LA1 data set

Hyperclique patterns	Support	H-confidence
{gorbachev, mikhail}	1.4%	93.6%
{photo, graphic, writer}	14.5%	42.1%
{sentence, convict, prison}	1.4%	32.4%
{rebound, score, basketball}	3.8%	40.2%
{season, team, game, play}	7.1%	31.4%

Union. Certainly, the presence of *mikhail* in one document strongly implies the presence of *gorbachev* in the same document and vice-versa.

HICAP: Hierarchical Clustering with Pattern Preservation

HICAP (Xiong et al., 2004) is based on the Group Average agglomerative hierarchical clustering technique, which is also known as UPGMA (Jain & Dubes, 1988). However, unlike the traditional version of UPGMA, which starts from clusters consisting of individual objects or attributes, HICAP uses hyperclique patterns to define the initial clusters, i.e., the objects of each hyperclique pattern become an initial cluster.

This algorithm consists of two phases. In phase I, HICAP finds maximal hyperclique patterns, which are the patterns we want to preserve in the HICAP algorithm. We use only maximal hyperclique patterns since any non-maximal hyperclique will, during the clustering process, tend to be absorbed by its corresponding maximal hyperclique pattern and will, therefore, not affect the clustering process significantly. Thus, the use of non-maximal hypercliques would add complexity without providing any compensating benefits.

In phase II, HICAP conducts hierarchical clustering and outputs the clustering results. We highlight several important points. First, since hyperclique patterns can be overlapping, some of the resulting clusters will be overlapping. Second, identified maximal hyperclique patterns typically cover only *10%* to *20%* of all objects, and thus, HICAP also includes each uncovered object as a separate initial cluster, i.e., the hierarchical clustering starts with maximal hyperclique patterns and uncovered objects. Finally, the similarity between clusters is calculated using the average of the pairwise similarities between objects, where the similarity between objects is computed using the cosine measure.

Hyperclique Patterns vs. Frequent Itemsets

As mentioned, there has been some prior work that uses frequent itemsets as the basis for clustering algorithms (Beil, Ester & Xu, 2002; Benjamin, Fung & Wang 2003; Wang, Xu & Liu, 1999). While the goals and methods of that work are different from our own, we could have used frequent itemsets instead of hypercliques in HICAP. We chose hypercliques because we feel that they have several advantages for pattern preserving clustering. First, since hyperclique patterns have the high affinity property; that is, they include objects that are strongly similar to each other with respect to cosine similarity and that should, therefore, naturally be within the same cluster. In contrast, many pairs of objects from a frequent itemset may have very poor similarity with respect to the cosine measure. Second, the hyperclique pattern mining algorithm has much better performance at low levels of support than frequent itemset mining algorithms. Thus, capturing patterns occurring at low levels of support is much easier for hyperclique patterns than frequent itemsets. Finally, the size of maximal hyperclique patterns is significantly smaller than the size of maximal frequent itemsets. In summary, a version of HICAP that uses hypercliques is far more computationally efficient that a version of HICAP that uses frequent patterns, and is more likely to produce meaningful clusters.

FUTURE TRENDS

While this chapter showed the potential usefulness of pattern preserving clustering based on a hierarchical clustering approach, more work is necessary to show the broad applicability and usefulness of a pattern preserving approach to clustering. We also hope to investigate how pattern preserving clustering might be incorporated into non-hierarchical clustering schemes.

Hierarchical Clustering with Pattern Preservation can be made more general by focusing on patterns other than hyperclique patterns. We need to investigate what conditions these patterns must meet for this to be meaningful. In particular, do patterns need to possess the high affinity property? More generally, we hope to understand what must be done to modify clustering techniques to be compatible with the goal of preserving patterns.

CONCLUSIONS

In this chapter, we have introduced a new goal for clustering algorithms, namely, the preservation of patterns, such as the hyperclique pattern, that capture strong connections between objects. Without such an explicit goal, clustering algorithms tend to find clusters that split the objects in these patterns into different

clusters. However, keeping these patterns together aids cluster interpretation.

REFERENCES

Agrawal, R., Imielinski, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. In *Proceedings of the ACM SIGMOD*, 207-216.

Anderberg, M. R. (1973). *Cluster Analysis for Applications*. New York: Academic Press

Beil, F., Ester, M., & Xu, X. (2002). Frequent term-based text clustering. In *Proceedings of the ACM SIGKDD*, 436-442.

Fung, B. C. M., Wang, K., & Ester, M. (2003). Hierarchical document clustering using frequent itemsets. In *Proceedings of SIAM International Conference on Data Mining*.

Berkhin, P. (2002). Survey of clustering data mining techniques. Technical report, Accrue Software, CA.

Davidson, I., & Ravi, S. S. (2005). Clustering under Constraints: Feasibility Results and the K-Means Algorithm, In *Proceedings of SIAM International Conference on Data Mining*.

Han, J., Kamber, M., & Tung, A. K. H. (2001). Spatial clustering methods in data mining: A review. In Miller, H. J., & Han, J., editors, *Geographic Data Mining and Knowledge Discovery*, 188-217. London. Taylor and Francis.

Jain, A. K., & Dubes, R. C. (1988). *Algorithms for Clustering Data*. Prentice Hall Advanced Reference Series. Englewood Cliffs, New Jersey. Prentice Hall.

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: A review. *ACM Computing Surveys*, **31**(3), 264-323.

Kaufman, L., & Rousseeuw, P. J. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*. New York: Wiley Series in Probability and Statistics. John Wiley and Sons.

MacQueen, J. (1999). Some methods for classification and analysis of multivariate observations. In L. M. L. Cam and J. Neyman, editors, *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume I, Statistics*. University of California Press.

Tung, A. K. H., Ng, R. T., Lakshmanan, L. V. S., & Han, J. (2001). Constraint-based clustering in large databases. In den Bussche, J. V. & Vianu, V., editors, *Database Theory - ICDT 2001*.

Wagstaff, K., & Cardie, C. (2000). Clustering with Instance- Level Constraints, In *Proceedings of 17th Intl. Conf. on Machine Learning (ICML 2000)*, Stanford, CA, 1103-1110.

Wang, K., Xu, C., & Liu, B. (1999). Clustering transactions using large items. In *Proceedings of ACM International Conference on Information and Knowledge Management*, 483-490.

Xiong, H., He, X., Ding, C. H. Q., Zhang, Y., Kumar, V., & Holbrook, S. R. (2005). Identification of Functional Modules in Protein Complexes via Hyperclique Pattern Discovery. *Pacific Symposium on Biocomputing*, 221-232

Xiong, H., Steinbach, M., Tan, P.-N., & Kumar, V. (2004). HICAP: Hierarchical Clustering with Pattern Preservation. In *Proc. of SIAM International Conference on Data Mining*, 279 - 290

Xiong, H., Tan, P.-N., & Kumar, V. (2003). Mining strong affinity association patterns in data sets with skewed support distribution. In *Proceedings of the 3rd IEEE International Conference on Data Mining*, 387-394.

Xiong, H., Tan, P.-N., & Kumar, V. (2006). Hyperclique pattern discovery. *Data Mining and Knowledge Discovery*, **13**(2), 219-242.

KEY TERMS

Clustering Analysis: The process of dividing objects into groups (clusters) in a way such that objects in a cluster are more similar to each other than to objects in other clusters.

Confidence: The confidence of an association rule $X_1 \rightarrow X_2$ is defined as $conf(X_1 \rightarrow X_2) = supp(X_1 \cup X_2)/supp(X_1)$. It is the fraction of the transactions containing the items in X_1 that also contain the items

in X_2, and is an estimate of the conditional probability of X_2 given X_1.

H-Confidence: The h-confidence of an itemset $P = \{i_1, i_2, ..., i_m\}$ is defined as $hconf(P) = min_{1 \le k \le m}\{conf(i_k \rightarrow P - i_k)\}$.

Hierarchical Clustering: It is a clustering approach which produces a nested series of partitions of the data.

Hyperclique Pattern: Given a set of items I and a minimum h-confidence threshold h_c, an itemset $P \subseteq I$ is a hyperclique pattern if and only if $hconf(P) \ge h_c$.

Itemset: Let $I = \{i_1, i_2, ..., i_m\}$ be a set of distinct items. Each transaction T in database D is a subset of I. We call $X \subseteq I$ an itemset.

Maximal Hyperclique Pattern: A hyperclique pattern is a maximal hyperclique pattern if no superset of this pattern is a hyperclique pattern.

Pattern Preserving Clustering: It is a clustering approach which preserves meaningful and interpretable patterns during the clustering process. The patterns preserved in clusters can help interpret the clustering results.

Support: The support of an itemset X, denoted by $supp(X)$, is the fraction of transactions containing the itemset X.

Pattern Synthesis for Nonparametric Pattern Recognition

P. Viswanath
Indian Institute of Technology-Guwahati, India

M. Narasimha Murty
Indian Institute of Science, India

Shalabh Bhatnagar
Indian Institute of Science, India

INTRODUCTION

Parametric methods first choose the form of the model or hypotheses and estimates the necessary parameters from the given dataset. The form, which is chosen, based on experience or domain knowledge, often, need not be the same thing as that which actually exists (Duda, Hart & Stork, 2000). Further, apart from being highly error-prone, this type of methods shows very poor adaptability for dynamically changing datasets. On the other hand, non-parametric pattern recognition methods are attractive because they do not derive any model, but works with the given dataset directly. These methods are highly adaptive for dynamically changing datasets. Two widely used non-parametric pattern recognition methods are (a) the nearest neighbor based classification and (b) the Parzen-Window based density estimation (Duda, Hart & Stork, 2000). Two major problems in applying the non-parametric methods, especially, with large and high dimensional datasets are (a) the high computational requirements and (b) the curse of dimensionality (Duda, Hart & Stork, 2000). Algorithmic improvements, approximate methods can solve the first problem whereas feature selection (Isabelle Guyon & André Elisseeff, 2003), feature extraction (Terabe, Washio, Motoda, Katai & Sawaragi, 2002) and bootstrapping techniques (Efron, 1979; Hamamoto, Uchimura & Tomita, 1997) can tackle the second problem. We propose a novel and unified solution for these problems by deriving a *compact and generalized abstraction* of the data. By this term, we mean a compact representation of the given patterns from which one can retrieve not only the original patterns but also some artificial patterns. The compactness of the abstraction reduces the computational requirements, and its generalization reduces the curse of dimensionality effect. Pattern synthesis techniques accompanied with compact representations attempt to derive compact and generalized abstractions of the data. These techniques are applied with (a) the nearest neighbor classifier (NNC) which is a popular non-parametric classifier used in many fields including data mining since its conception in the early fifties (Dasarathy, 2002) and (b) the Parzen-Window based density estimation which is a well known non-parametric density estimation method (Duda, Hart & Stork, 2000).

BACKGROUND

Pattern synthesis techniques, compact representations and its application with NNC and Parzen-Window based density estimation are based on more established fields:

- **Pattern recognition:** Statistical techniques, parametric and non-parametric methods, classifier design, nearest neighbor classification, probability density estimation, curse of dimensionality, similarity measures, feature selection, feature extraction, prototype selection, and clustering techniques.
- **Data structures and algorithms:** Computational requirements, compact storage structures, efficient nearest neighbor search techniques, approximate search methods, algorithmic paradigms, and divide-and-conquer approaches.
- **Database management:** Relational operators, projection, cartesian product, data structures, data management, queries, and indexing techniques.

MAIN FOCUS

Pattern synthesis, compact representations followed by its application with NNC and Parzen-Window density estimation are described below.

Pattern Synthesis

Generation of artificial new patterns using the given set of patterns is called pattern synthesis. Instance based pattern synthesis uses the given training patterns and some of the properties of the data. It can generate a finite number of new patterns. Computationally this can be less expensive than deriving a model from which the new patterns can be extracted. This is especially useful for non-parametric methods like NNC and Parzen-Window based density estimation (Duda, Hart and Stork, 2000) which directly use the training instances. It is argued that using a larger synthetic set can reduce the bias of the density estimation or classification (Viswanath, Murty & Bhatnagar, 2006). Further, the usage of the respective compact representations can result in reduction of the computational requirements.

This chapter presents two instance based pattern synthesis techniques called *overlap based pattern synthesis* and *partition based pattern synthesis* and their corresponding compact representations.

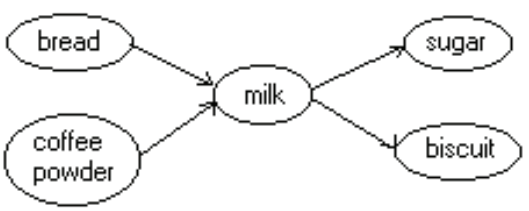
Overlap Based Pattern Synthesis

Let F be the set of features (or attributes). There may exist a three-block partition of F, say, $\{A, B, C\}$ with the following properties. For a given class, there is a dependency (probabilistic) among features in A U B. Similarly, features in B U C have a dependency. However, features in A (or C) can affect those in C (or A) only through features in B. That is, to state more formally, A and C are statistically independent given B. Suppose that this is the case and we are given two patterns $X = (a_1, b, c_1)$ and Y = (a_2, b, c_2) such that a_1 is a feature-vector that can be assigned to the features in A, b to the features in B and c_1 to the features in C, respectively. Similarly, a_2, b and c_2 are feature-vectors that can be assigned to features in A, B, and C, respectively. Then, our argument is that the two patterns (a_1, b, c_2) and (a_2, b, c_1) are also valid patterns in the same class or category as X and Y. If these two new patterns are not already in the class of patterns then it is only because of the finite nature of the set. We call this type of generation of additional patterns as *overlap based pattern synthesis*, because this kind of synthesis is possible only if the two given patterns have the same feature-values for features in B. In the given example, feature-vector b is common between X and Y and therefore is called the *overlap*. This method is suitable only with discrete valued features (can be of symbolic or categorical types also). If more than one such partition exists then the synthesis technique is applied sequentially with respect to the partitions in some order.

One simple example to illustrate this concept is as follows. Consider a supermarket sales database where two records, *(bread, milk, sugar)* and *(coffee, milk, biscuits)* are given. Let us assume, it is known that there is a dependency between (i) *bread* and *milk*, (ii) *milk* and *sugar*, (iii) *coffee* and *milk*, and (iv) *milk* and *biscuits*. Then the two new records that can be synthesized are *(bread, milk, biscuits)* and *(coffee, milk, sugar)*. Here *milk* is the overlap. A compact representation in this case is shown in Figure 1 where a path from left to right ends denotes a data item or pattern. So we get four patterns in total from the graph shown in Figure 1 (two original and two synthetic patterns). Association rules derived from association rule mining (Han & Kamber, 2001) can be used to find these kinds of dependencies. Generalization of this concept and its compact representation for large datasets are described below.

If the set of features, F can be arranged in an order such that $F = \{f_1, f_2, ..., f_d\}$ is an ordered set with f_k being the k^{th} feature and all possible three-block partitions can be represented as $P_i = \{A_i, B_i, C_i\}$ such that $A_i = (f_1, ..., f_a)$, $B_i = (f_{a+1}, ..., f_b)$ and $C_i = (f_{b+1}, ..., f_d)$ then the compact representation called *overlap pattern graph* is described with the help of an example.

Figure 1.

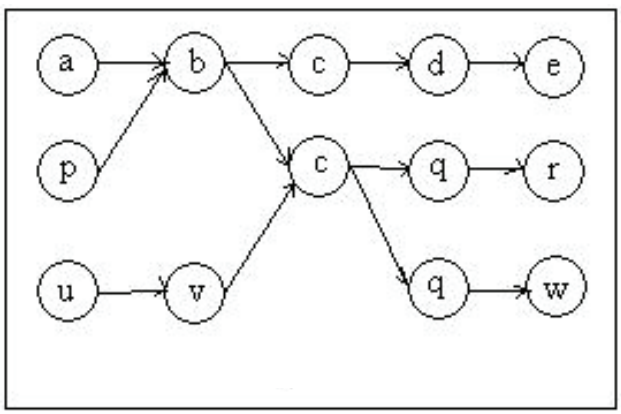


Overlap Pattern Graph (OLP-graph)

Let $F = (f_1, f_2, f_3, f_4, f_5)$. Let two partitions satisfying the conditional independence requirement be $P_1 = \{ \{f_1\}, \{f_2, f_3\}, \{f_4, f_5\}\}$ and $P_2 = \{ \{f_1, f_2\}, \{f_3, f_4\}, \{f_5\}\}$. Let three given patterns be *(a,b,c,d,e), (p,b,c,q,r)* and *(u,v,c,q,w)*, respectively. Since *(b,c)* is common between the1st and 2nd patterns, two synthetic patterns that can be generated are *(a,b,c,q,r)* and *(p,b,c,d,e)*. Likewise three other synthetic patterns that can be generated are *(p,b,c,d,e), (p,b,c,q,w)* and *(a,b,c,q,w)* (note that, the last synthetic pattern is derived from two earlier synthetic patterns). A compact representation called *overlap pattern graph (OLP-graph)* for the entire set (including both given and synthetic patterns) is shown in Figure 2 where a path from left end to right end represents a pattern. The graph is constructed by inserting the given patterns, whereas the patterns that can be extracted out of the graph form the entire synthetic set consisting of both original and synthetic patterns. Thus from the graph given in Figure 2, a total of eight patterns can be extracted, out of which five are new synthetic patterns.

OLP-graph can be constructed by scanning the given dataset only once and is independent of the order in which the given patterns are considered. An approximate method for finding partitions, a method for construction of OLP-graph and its application to NNC are described in (Viswanath, Murty & Bhatnagar, 2005). For large datasets, this representation reduces the space requirement drastically.

Figure 2.



Partition Based Pattern Synthesis

Let $P = \{A1, A2,...,Ak\}$ be a k block partition of F such that $A1, A2,...,Ak$ are statistically independent subsets. Let $\mathbf{Y}$ be the given dataset and $Proj_{A1}(\mathbf{Y})$ be the projection of $\mathbf{Y}$ onto features in the subset $A1$. Then the cartesian product:

$$Proj_{A1}(\mathbf{Y}) \times Proj_{A2}(\mathbf{Y}) \times \ldots \times Proj_{Ak}(\mathbf{Y})$$

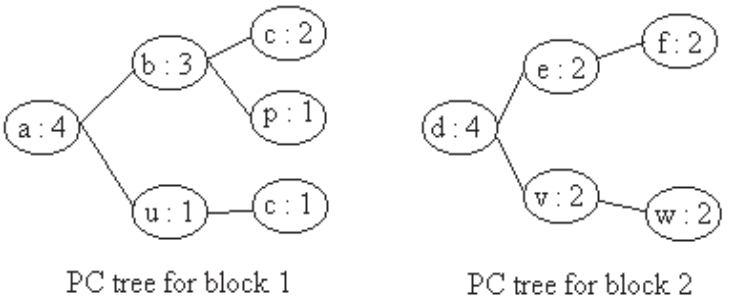
is the synthetic set generated using this method.

The partition P satisfying the above requirement can be obtained from domain knowledge or from association rule mining. Approximate partitioning methods using mutual information or pair-wise correlation between the features can also be used to get suitable partitions and are experimentally demonstrated to work well with NNC in (Viswanath, Murty & Bhatnagar, 2004).

Partitioned Pattern Count Tree (PPC-Tree)

This compact representation, suitable for storing the partition based synthetic patterns, is based on Pattern Count Tree (PC-tree) (Ananthanarayana, Murty & Subramanian, 2001) which is a prefix tree like structure. PPC-tree is a generalization of PC-tree such that a PC-tree is built for each of the projected patterns i.e., $Proj_{Ai}(\mathbf{Y})$ for I = 1 to k. PPC-tree is a more compact structure than PC-tree which can be built by scanning the dataset only once and is independent of the order in which the given patterns are considered. The construction method and properties of PPC-tree can be found in (Viswanath, Murty & Bhatnagar, 2004). As an example, consider four given patterns *viz., (a,b,c,d,e,f), (a,u,c,d,v,w), (a,b,p,d,e,f), (a,b,c,d,v,w)* and a two block partition where the first block consists of first three features and the second block consists of remaining three features. The corresponding PPC-tree is shown in Figure 3 where each node contains a *feature-value* and an integer called *count* which gives the number of patterns sharing that node. There are two PC-trees, one corresponding to each block. A path in the PC-tree for block 1 (from root to leaf) concatenated with a path in the PC-tree for block 2 gives a pattern that can be extracted. So, in total six patterns can be extracted from the structure shown in Figure 3.

Figure 3.



An Application of Synthetic Patterns with Nearest Neighbor Classifier and Parzen-Window Based Density Estimation

Classification and prediction methods are among various elements of data mining and knowledge discovery such as association rule mining, clustering, link analysis, rule induction, etc (Wang, 2003). The nearest neighbor classifier (NNC) is a very popular non-parametric classifier (Dasarathy, 2002). It is widely used in various fields because of its simplicity and good performance. Theoretically, with infinite number of samples its error rate is upper-bounded by twice the error of Bayes classifier (Duda, Hart, & Stork, 2000). NNC is a lazy classifier in the sense that no general model (like a decision tree) is built until a new sample needs to be classified. So, NNC can easily adapt to situations where the dataset changes frequently. But, computational requirements and curse of dimensionality are the two major issues that need to be addressed in order to make the classifier suitable for data mining applications (Dasarathy, 2002). Prototype selection (Susheela Devi & Murty, 2002), feature selection (Isabelle Guyon & Andre Elisseeff, 2003), feature extraction (Terabe, Washio, Motoda, Katai & Sawaragi, 2002), compact representations of the datasets (Maleq Khan, Qin Ding & William Perrizo, 2002) and bootstrapping (Efron, 1979 and Hamamoto, Uchimura & Tomita, 1997) are some of the remedies to tackle these problems. But all these techniques need to be followed one after the other, *i.e.*, they cannot be combined together. Pattern synthesis techniques and compact representations described in this chapter provide a unified solution. Similar to NNC, the Parzen-Window based density estimation is a popular non-parametric density estimation method (Viswanath, Murty, & Kambala, 2005).

Efficient implementations of NNC which can directly work with OLP-graph are given in (Viswanath, Murty & Bhatnagar, 2005). These use dynamic programming techniques to reuse the partial distance computations and thus reduce the classification time requirement. Partition based pattern synthesis is presented in along with efficient NNC methods (Viswanath, Murty, & Bhatnagar, 2006). An *efficient NNC implementation with constant classification time* is presented in (Viswanath, Murty, & Bhatnagar, 2004). Pattern synthesis and compact representation schemes can reduce the bias of the density estimation and are shown to perform well in a density based intrusion detection system (Viswanath, Murty, & Kambala, 2005). These methods are, in general, based on the divide-and-conquer approach and are efficient to work directly with the compact representation. Thus, the computational requirements are reduced. Since the total number of patterns (i.e., the size of the synthetic set) considered is much larger than those in the given training set, the effect of curse of dimensionality is reduced.

FUTURE TRENDS

Integration of various data mining tasks like association rule mining, feature selection, feature extraction, classification, etc., using compact and generalized abstractions is a very promising direction. Apart from NNC and Parzen-Window based density estimation methods, other applications using synthetic patterns and corresponding compact representations like decision tree induction, rule deduction, clustering, etc., are also interesting areas which can give valuable results.

CONCLUSION

Pattern synthesis is a novel technique which can enhance the generalization ability of lazy learning methods like NNC and Parzen-Window based density estimation by reducing the curse of dimensionality effect and also the corresponding compact representations can reduce the computational requirements. In essence, it derives a compact and generalized abstraction of the given dataset. Overlap based pattern synthesis and partition based pattern synthesis are two instance based pattern synthesis methods where OLP-graph, PPC-tree are respective compact storage structures.

REFERENCES

Ananthanarayana, V.S., Murty, M.N,. & Subramanian, D.K. (2001). An incremental data mining algorithm for compact realization of prototypes, *Pattern Recognition 34*, 2249-2251.

Dasarathy, B.V. (2002). Data mining tasks and methods: Classification: Nearest-neighbor approaches. In *Handbook of data mining and knowledge discovery*, 288-298. New York: Oxford University Press.

Duda, R.O, Hart, P.E & Stork, D.G. (2000). *Pattern classification, 2nd Edition*. Wiley Interscience Publication.

Efron, B. (1979) Bootstrap methods: Another look at the Jackknife. *Annual Statistics, 7*, 1-26.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, *3*(March),1157-1182.

Hamamoto, Y., Uchimura, S., & Tomita, S. (1997). A bootstrap technique for nearest neighbor classifier design. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*(1), 73-79.

Han, J., & Kamber, M. (2001). *Data mining: Concepts and techniques*. Morgan Kaufmann Publishers.

Khan, M., Ding, Q., & Perrizo, W. (2002). K-nearest neighbor classification on spacial data streams using p-trees. *PAKDD 2002*, LNAI 336, 517-528. Springer-Verlag.

Susheela Devi, V., & Murty, M.N. (2002). An incremental prototype set building technique. *Pattern Recognition Journal*, *35*, 505-513.

Terabe, M., Washio, T., Motoda, H., Katai, O., & Sawaragi, T. (2002). Attribute generation based on association rules. *Knowledge and Information Systems*, *4*(3), 329-349.

Viswanath P., Murthy, M.N, & Bhatnagar, S. (2004). Fusion of multiple approximate nearest neighbor classifiers for fast and efficient classification, *Information Fusion Journal*, *5*, 239-250.

Viswanath, P., Murty, M.N, & Bhatnagar, S. (2005). Overlap based pattern synthesis with an efficient nearest neighbor classifier. *Pattern Recognition Journal 38*, 1187-1195.

Viswanath, P., Murty, M.N, & Kambala, S. (2005). An efficient Parzen-window based network intrusion detector using a pattern synthesis technique. *Proceedings of the First International Conference on Pattern Recognition and Machine Intelligence*, Indian Statistical Institute, Kolkata, 2005, 799-804.

Viswanath, P., Murty, M.N, & Bhatnagar, S. (2006). Partition based pattern synthesis with efficient algorithms for nearest neighbor classification. *Pattern Recognition Letters*, *27*, 1714-1724.

Wang, J. (2003). *Data mining: Opportunities and challenges*. Hershey, PA: Idea Group Publishing.

KEY TERMS

Bootstrapping: Generating artificial patterns from the given original patterns. (This does not mean that the artificial set is larger in size than the original set, also artificial patterns need not be necessarily distinct from the original patterns.)

Curse of Dimensionality Effect: The number of samples needed to estimate a function (or model) grows exponentially with the dimensionality of the data.

Compact and Generalized Abstraction of the Training Set: A compact representation built by using the training set from which not only the original patterns but some of new synthetic patterns can also be derived.

Prototype Selection: The process of selecting a few representative samples from the given training set suitable for the given task.

Training Set: The set of patterns whose class labels are known and which are used by the classifier in classifying a given pattern.

Pattern Synthesis in SVM Based Classifier

C. Radha
Indian Institute of Science, India

M. Narasimha Murty
Indian Institute of Science, India

INTRODUCTION

An important problem in pattern recognition is that of pattern classification. The objective of classification is to determine a discriminant function which is consistent with the given training examples and performs reasonably well on an unlabeled test set of examples. The degree of performance of the classifier on the test examples, known as its generalization performance, is an important issue in the design of the classifier. It has been established that a good generalization performance can be achieved by providing the learner with a sufficiently large number of discriminative training examples. However, in many domains, it is infeasible or expensive to obtain a sufficiently large training set. Various mechanisms have been proposed in literature to combat this problem. Active Learning techniques (Angluin, 1998; Seung, Opper, & Sompolinsky, 1992) reduce the number of training examples required by carefully choosing discriminative training examples. Bootstrapping (Efron, 1979; Hamamoto, Uchimura & Tomita, 1997) and other pattern synthesis techniques generate a synthetic training set from the given training set. We present some of these techniques and propose some general mechanisms for pattern synthesis.

BACKGROUND

Generalization performance is generally quantified using the technique of structural risk minimization (Vapnik, 1998). The risk of misclassification is dependent on the training error, the V-C dimension of the classifier and the number of training examples available to the classifier. A good classifier is one which has a low training error and low V-C dimension. The generalization performance of the classifier can also be improved by providing a large number of training examples.

The number of training examples required for efficient learning depends on the number of features in each training example. Larger the number of features, larger is the number of training examples required. This is known as the curse of dimensionality. It is generally accepted that at least ten times as many training examples per class as the number of features are required (Jain & Chandrasekharan, 1982).

However, in most applications, it is not possible to obtain such a large set of training examples. Examples of such domains are:

1. **Automated Medical diagnosis:** This involves designing a classifier which examines medical reports and determines the presence or absence of a particular disorder. For efficient classification, the classifier must be trained with a large number of positive and negative medical reports. However, an adequate number of medical reports may not be available for training because of the possible high cost of performing the medical tests.
2. **Spam Filtering:** A good mail system aims at identifying and filtering out spam mails with minimum help from the user. The user labels mails as spam as and when she encounters them. These labeled mails are used to train a classifier which can further be used to identify spam mails. For the mail system to be useful, the user should be presented with the least possible number of spam mails, which calls for an efficient classifier. However, sufficient training examples are not available to the learner to perform efficient classification.
3. **Web page recommendations:** The user is provided recommendations based on her browsing history. If good recommendations are to be provided early, then the recommendation system will have to manage with a relatively low number of

examples representing web pages of interest to the user.

There are two major techniques that can be employed to combat the lack of a sufficiently large training set: Active Learning and Pattern Synthesis.

ACTIVE LEARNING

Active Learning is a technique in which the learner exercises some control over the training set. In pool-based active learning, a learner is provided with a small set of labeled training examples and a large set of unlabeled examples. The learner iteratively chooses a few examples from the unlabeled examples for labeling; the user provides the labels for these examples and the learner adds these newly labeled examples to its training set.

In the "membership query" paradigm of active learning, the learner queries the user for the label of a point in the input region (Angluin, 1998).

Active Learning aims at finding a good discriminant function with minimum number of labeled training examples. The version space for a given set of training examples is defined as the set of all discriminant functions that consistently classify the training examples. Version space is the region of uncertainty, the region which contains those examples whose labels the learner is uncertain about. Active Learning methods choose from this region, those unlabeled examples for labeling, which reduce the version space as fast as possible, ideally by half in each iteration.

Various algorithms for active learning have been proposed in literature. Some of them are "Query by Committee" (Seung, Opper, & Sompolinsky, 1992), "Committee Based Sampling" (Dagan, & Engelson, 1995), etc. These techniques train a committee of classifiers and after each iteration of training, the labels of the unlabeled examples are determined individually by each member of the committee. That unlabeled example for which there is maximum disagreement among the committee members is chosen for labeling. Active Learning has been incorporated along with the Expectation – Maximization algorithm to improve efficiency (Nigam, McCallum, Thrun, & Mitchell, 2000). This technique has also been employed in SVM training (Tong & Koller, 2001; Tong & Chang, 2001).

PATTERN SYNTHESIS

In some domains it may not be possible to obtain even unlabeled examples. In some others, the user may not be available or may not be in a position to provide the labels for the unlabeled examples. In especially such circumstances, it is useful to generate artificial training patterns from the given training set. This process is known as pattern synthesis.

Various techniques have been devised in literature to obtain an artificial training set. Some of the techniques are described below in brief:

1. **Bootstrapping:** This technique involves approximating the sample probability distribution from the given examples and drawing random samples from this distribution. This sample is called the bootstrap sample and the distribution is called the bootstrap distribution. The bootstrap distribution can be estimated through various means, including direct calculation, Monte Carlo estimation and Taylor's series expansion (Efron, 1979). This technique has been successfully applied in Nearest Neighbor Classifier design (Hamamoto, Uchimura & Tomita, 1997).
2. **Using Domain-related information:** These techniques involve exploiting domain knowledge such as radial symmetry (Girosi & Chan, 1995) and transformation invariance (Niyogi, Girosi & Poggio, 1998). Such invariances have been applied to the support vectors obtained from a Support Vector Machine (Scholkopf, Burges & Vapnik, 1996). In this technique, an initial model is learnt to determine the support vectors in the training set. The transformations are then applied to these support vectors to obtain synthetic patterns.

MAIN FOCUS

General Pattern Synthesis Techniques

These techniques involve applying a transformation to the given training examples to obtain new training

examples. The transformation applied should ensure that the generated examples lie in the same class region as their corresponding parent examples. Finding such a transformation would be simple if the actual class distributions were known apriori. However, this information is not available in practical classification problems. If this information was available, it could be used to learn the discriminant directly instead of generating training examples. Thus obtaining such a transformation is in itself an important problem.

The determination of the transformation can be formulated as the following optimization problem: Let $X = \{x_k, y_k\}_{k=1}^{N}$ be the available training set where x_k are vectors in $\Re^n$ and $y_k \in \{-1,1\}$ are the corresponding class labels. We synthesize a set of patterns X' using X by applying a transformation T to it. Let J be the optimizing criterion for the classifier, $f(x)$ be the discriminant function learnt when only X is provided as the training set to the learner and $g(x)$ be the discriminant function learnt when $X \cup X'$ is provided for training.

$$\begin{aligned} &min \quad J \\ subject\ to \quad &f(x_k)g(T(x_k)) \geq 0 \\ &y_k g(T(x_k)) \geq 0 \\ &y_k g(x_k) \geq 0 \quad k = 1,2,...,N \end{aligned}$$

The first constraint ensures that the parent pattern from X and the corresponding synthesized pattern in X' are given the same label by the classifier. The second and third constraints ensure that $g(x)$ consistently classifies patterns belonging to both X and X'.

We examine two different types of transformations that can be applied to the training set to obtain the artificial training set:

- Scaling which uses the transformation $T(x) = \gamma x$.
- Perturbation which uses the transformation $T(x) = x + \gamma$.

In both these methods the value of γ should be chosen such that all the constraints in the above optimization problem are satisfied.

Pattern Synthesis in Support Vector Machines

Support Vector Machines (SVM) (Burges, 1998; Sastry, 2003) find a hyperplane which separates the training examples with the largest possible margin. If the training examples are not linearly separable, the training examples are projected into a higher dimensional space where they are more likely to be linearly separable. The separating hyperplane in this higher dimension translates to a non-linear discriminant in the original input space. SVMs achieve a good generalization performance by minimizing both the training error and the V-C dimension.

A Δ margin separating hyperplane is one which classifies a test example x using the following decision rule:

if $w^T x + b > \Delta$ class $y = +1$
if $w^T x + b \leq \Delta$ class $y = -1$

The V-C dimension (Vapnik, 1998) of such a Δ margin separating hyperplane classifier in $\Re^n$ is bounded above as follows:

$$h \leq \min\left(\left[\frac{\rho^2}{\Delta^2}\right], n\right) + 1$$

where ρ is the radius of the sphere which encloses all the training examples. SVMs find the classifier that maximizes the margin Δ and hence has a low V-C dimension.

However, the SVM classifier may perform badly when the right training examples are not available. Consider the situation shown in Figure 1.

In order to classify the noisy examples correctly, the input examples may be projected to a higher dimensional space and the SVM may find a non-linear discriminant as shown in Figure 2.

This non-linear classifier is more complex than a simple hyperplane which ignores the noisy examples (Figure 3). This results in the deterioration of the performance of the classifier. Through pattern synthesis it is possible to obtain sufficient number of training examples which truly represent the class distributions, hence reducing the effect of such noisy patterns and improving the generalization performance of the classifier.

Figure 1.

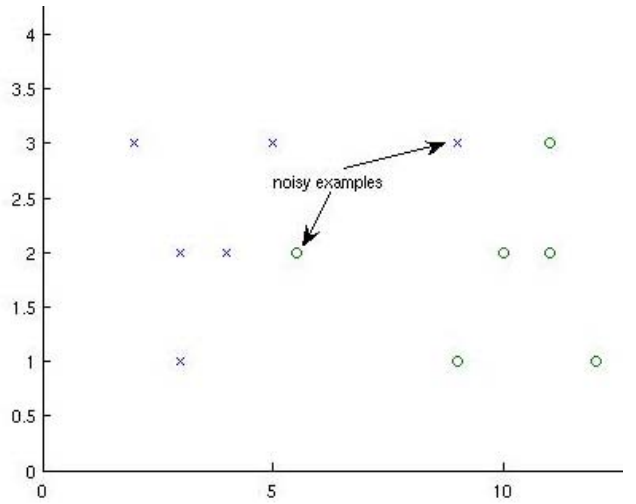


Figure 2.

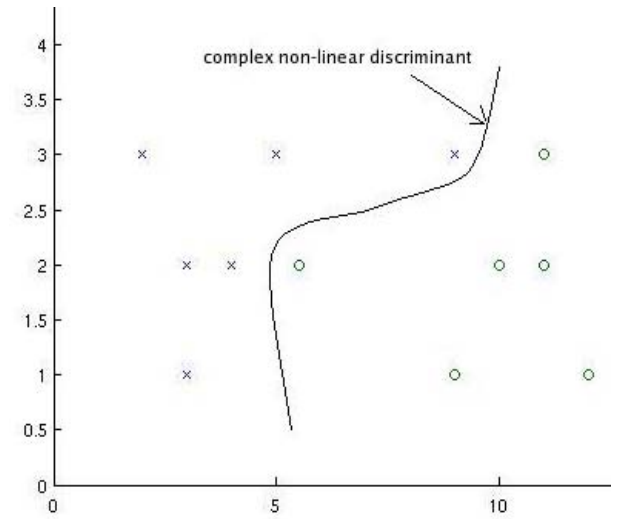
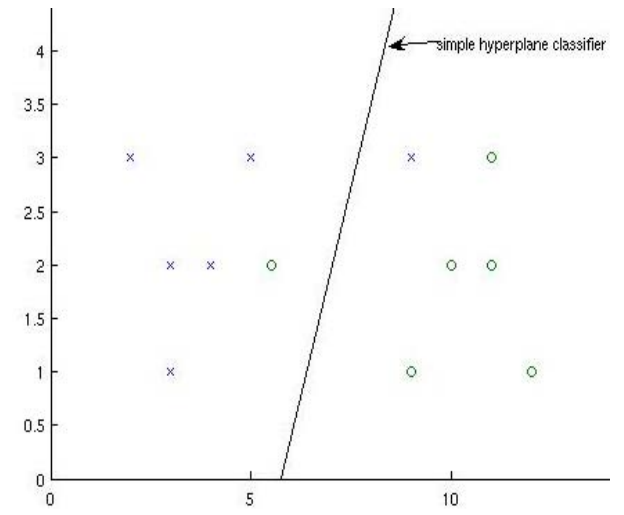


Figure 3.



In an SVM, the most important patterns are the Support Vectors, the patterns closest to the separating hyperplane. Hence, it is more meaningful to use these Support Vectors for pattern synthesis. We describe two algorithms based on the scaling and perturbation techniques discussed above, in the context of SVMs.

PATTERN SYNTHESIS THROUGH SCALING

Pattern Synthesis is performed as per the following algorithm: Using the given set of training patterns train an SVM and determine its support vectors. Scale each support vector pattern x_i by a factor γ_i chosen uniformly at random from the range $[0,\gamma]$ where γ is a user-defined parameter. Add the scaled patterns to the training set and repeat the training and synthesis process for a fixed number of iterations.

For example, again consider the situation in Figure 1. During the first training phase, the two noisy examples become support vectors. The feature vectors of these two examples are X_1 = (5.5, 2) and X_2 = (9, 3). By scaling X_1 by a factor of 2, we obtain X'_1 = (11, 4) which falls well into the region of the positive examples (marked as circles). Similarly, by scaling X_2 by a factor of 0.5, we obtain X'_2 = (4.5, 1.5) which falls well into the region of the negative examples (marked as crosses). Figure 4 illustrates the effect of such scaling.

Since the scaling factor for each example is different, each pattern shifts by a small amount producing an overall shift in the separating hyperplane. As the process of shifting is repeated a large number of times with random scaling factors, there is a high probability that the final separating hyperplane obtained truly separates the class regions.

Pattern Synthesis through Perturbation

Perturbation is used for pattern synthesis as follows: Determine the support vectors of the SVM trained using the given training patterns. For each of these support vector patterns, choose $m = n * p$ features to perturb. Here p is the perturbation probability, a user input and n is the dimension of the training patterns. Perturb each selected feature f_j by δ_j chosen uniformly at random in the range $[0, \Delta]$ where Δ is the maximum allowed perturbation, again provided by the user. Add the perturbed pattern to the training set. Repeat the

Figure 4.

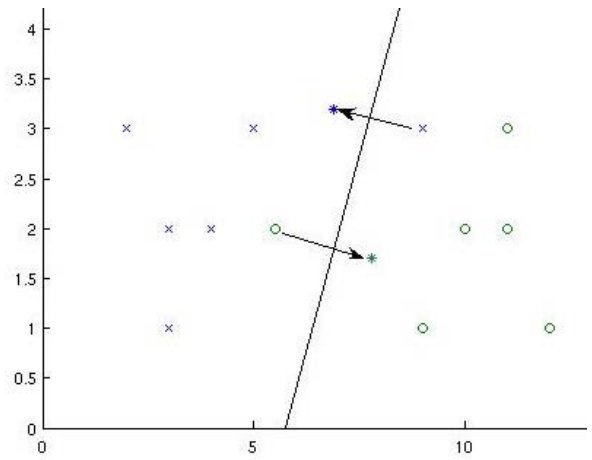


Figure 5.

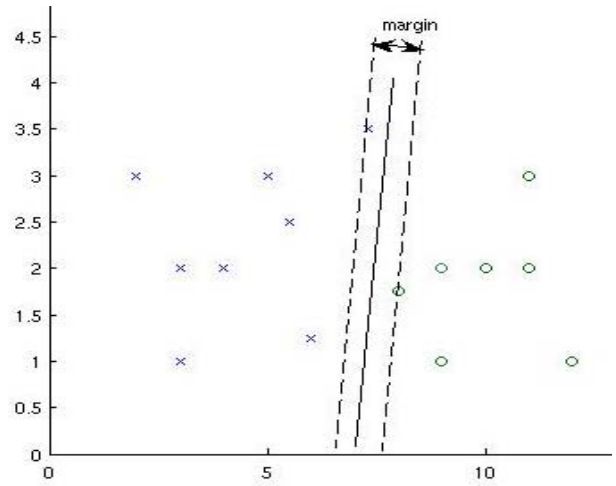
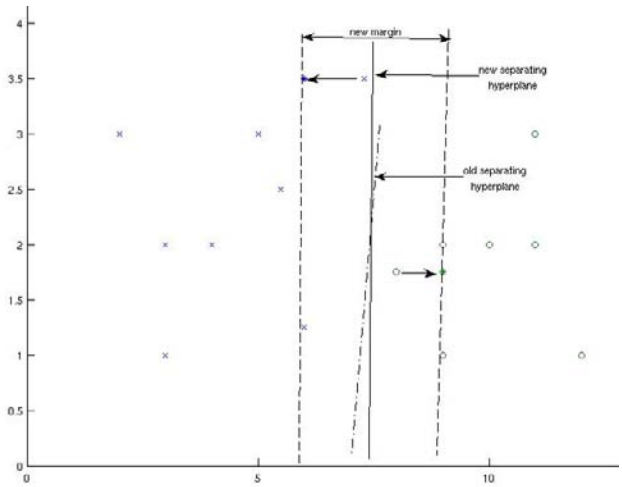


Figure 6.



training and synthesis processes for a fixed number of iterations. The value of Δ can be decided either using domain knowledge or based on the training patterns.

The generated pattern lies in the vicinity of the parent pattern and hence we may be assured that the synthetic pattern and its corresponding parent pattern have the same class label.

The perturbation method is particularly useful when small shifts are sufficient to arrive at the true separating hyperplane. Consider the situation in Figure 5.

SVM finds a separating hyperplane of a very small margin due to the possibly noisy patterns lying on the supporting hyperplanes. If these patterns were shifted slightly, a hyperplane with a larger margin can be found as shown in Figure 6. Larger margin implies lower V-C dimension which in turn implies better generalization performance.

FUTURE TRENDS

The pattern synthesis techniques proposed could be applied in other learning problems such as regression, clustering and association rule mining where a large and discriminative training set is useful. In the context of classification, the performance of these techniques with other classification algorithms such as Nearest Neighbor, Decision trees, etc. is to be verified.

CONCLUSION

This chapter presents some general techniques to synthesize an artificial training set which can be employed in addition to the original training set for training the classifier. The synthetic training set contributes significantly to the improvement in the generalization performance of the classifier.

In the context of pattern synthesis in SVMs, it can also be seen that the pattern synthesis technique induces an iterative nature into the training process. In each iteration, the training algorithm deals with a small subset of the entire training set, consisting only of the support vectors from the previous iteration. Therefore these techniques also contribute to reducing the time and space complexity of the training algorithm.

REFERENCES

Angluin,D.(1998). Queries and Concept Learning. *Machine Learning*, 2 (4),319-342.

Burges,C.J.C. (1998). A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 2, 121-167.

Dagan,I. & Engelson,S.P. (1995). Committee-based sampling for training probabilistic classifiers. *Twelfth International Conference on Machine Learning*, 150-157.

Efron,B. (1979). Bootstrap Methods: Another Look at JackKnife. *The Annals of Statistics*, 7, 1-26.

Girosi,F., & Chan,N. (1995). Prior knowledge and the creation of virtual examples for RBF networks. *IEEE-SP Workshop on Neural Networks for Signal Processing*, 201-210.

Hamamoto,Y., Uchimura,S. & Tomita,S. (1997). A bootstrap technique for nearest neighbor classifier design. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19, 73-79.

Jain,A.K. & Chandrasekharan,B. (1982). Dimensionality and sample size considerations in pattern recognition practice. *Handbook of Statistics*, 2, 835-855.

Nigam,K., McCallum,A.K., Thrun,S., & Mitchell,T. (2000). Text Classification from Labeled and Unlabeled Documents using EM. *Machine Learning*, 39 (2), 103-134.

Niyogi,P., Girosi,F., & Poggio,T. (1998). Incorporating prior information in machine learning by creating virtual examples. *Proceedings of IEEE*, 86, 2196-2209.

Sastry,P.S. (2003). An Introduction to Support Vector Machines. *Computing and Information Sciences: Recent Trends*, Editors: Goswami,A., Kumar,P., & Misra,J. C., Narosa Publishing House, New Delhi, 53-85.

Scholkopf,B., Burges,C. & Vapnik,V. (1996). Incorporating invariances in support vector learning machines. *Artificial Neural Networks*, 96, 47-52.

Seung,H.S., Opper,M., & Sompolinsky,H. (1992). Query by committee. *Fifth annual workshop on Computational Learning Theory*, 287-294.

Tong,S. & Chang,E. (2001). Support vector machine active learning for image retrieval. *Ninth ACM International Conference on Multimedia*, 107-118.

Tong,S. & Koller,D. (2001). Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 2 (1), 45-66.

Vapnik,V. N. (1998). *Statistical Learning Theory*. John Wiley & Sons, Inc.

KEY TERMS

Active Learning: A technique in which the learner carefully chooses the training examples with the objective of minimizing the number of training examples required for effective learning.

Bootstrapping: A technique for synthetic pattern generation which involves approximating the sample probability distribution from the given training examples and drawing random samples from this distribution.

Curse of Dimensionality: The number of training examples required to determine an accurate discriminant function grows exponentially with the dimensionality of the training data. This phenomenon is known as the curse of dimensionality.

Linear separability: A set of examples $\{x_k, y_k\}$ $k = 1, 2, ..., N$ where x_k are vectors in $\Re^n$ and $y_k \in \{-1,+1\}$ are said to be linearly separable if there exist $w \in \Re^n$ and $b \in \Re$ such that

$$w^T x_k + b > 0 \text{ if } y_k = +1 \qquad \text{and}$$
$$w^T x_k + b \leq 0 \text{ if } y_k = -1 \qquad \forall k = 1 \text{ to } N$$

Pattern Synthesis: Pattern synthesis may be defined as the generation of artificial patterns from a given set of patterns.

Risk of Misclassification: This is the loss incurred when a set of test examples are misclassified.

Support Vector Machine: Support vector machines are linear classifiers which map input vectors to a higher dimensional space where the training examples

become linearly separable and find a maximal separating hyperplane in this space.

Vapnik-Chervonenkis (V-C) dimension: This is a measure of the capacity of a class of classifiers. Let $F = \{f(x,a), a \in \Lambda\}$ be a set of discriminant functions which take values +1 or -1 and let X be a training set. A finite set $B = \{x_1, x_2, ..., x_m\} \subset X$ is said to be shattered by F if for any $C \subset B$, we can find a $\alpha \in \Lambda$ such that $f(x,a) = +1$ if $x \in C$ and $f(x,a) = +1$ if $x \in B - C$. The V-C dimension of F is defined to be the cardinality of the largest shattered subset of X.

The Personal Name Problem and a Data Mining Solution

Clifton Phua
Monash University, Australia

Vincent Lee
Monash University, Australia

Kate Smith-Miles
Deakin University, Australia

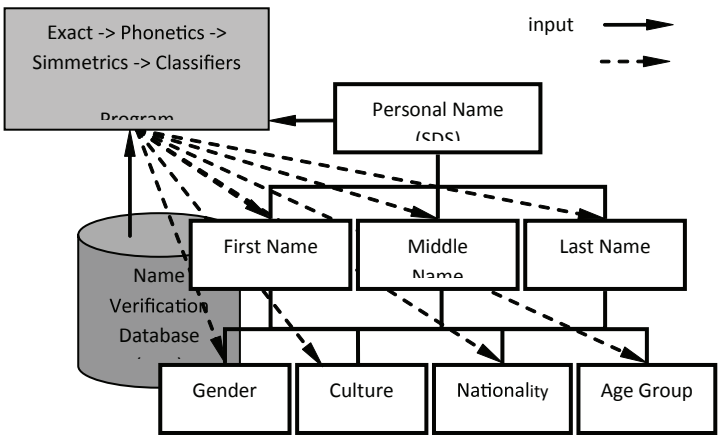
INTRODUCTION

Almost every person has a life-long personal name which is officially recognised and has only one correct version in their language. Each personal name typically has two components/parts: a first name (also known as given, fore, or Christian name) and a last name (also known as family name or surname). Both these name components are strongly influenced by cultural, economic, historical, political, and social backgrounds. In most cases, each of these two components can have more than a single word and the first name is usually gender-specific. (see Figure 1).

There are three important practical considerations for personal name analysis:

- Balance between manual checking and analytical computing. Intuitively, a small proportion of names should be manually reviewed, the result has to be reasonably accurate, and each personal name should not take too long to be processed.
- Reliability of the verification data has to be examined. By keeping the name verification database's updating process separate from incoming names, it can prevent possible data manipulation/corruption over time. However, the incompatibility of names in databases can also be caused by genuine reasons as such as cultural and historical traditions, translation and transliteration, reporting and recording variations, and typographical and phonetic errors (Borgman and Siegfried, 1992).

Figure 1. Hierarchy chart on the inputs, process, and outputs of the name verification task.



- Domain knowledge has to be incorporated into the entire process. Within the Australian context, the majority of names will be Anglo-Saxon but the minority will consist of many and very diverse groups of cultures and nationalities. Therefore the content of the name verification database has to include a significant number of popular Asian, African, Middle Eastern, and other names.

Figure 1 illustrates the input, process, and output sections. Input refers to the incoming names and those in the verification database (which acts like an external dictionary of legal names). Process/program refers to the possible four approaches for personal name analysis: exact-, phonetical-, similarity matching are existing and traditional approaches, while classification and hybrids are newer techniques on names-only data. Output refers to the insights correctly provided by the process. For simplicity, this paper uses first name to denote both first and middle names; and culture to represent culture and nationality. While the scope here explicitly seeks to extract first/last name and gender information from a personal name, culture can be inferred to a large extent (Levitt and Dubner, 2005), authenticity and age group can be inferred to a limited extent.

BACKGROUND

In this paper, we argue that there are four main explanations when the incoming first and last name does not match any name in the verification database exactly. First, the personal name is not authentic and should be manually checked. Second, it is most likely due to an incomplete white list. It is impossible to have a name verification database which has every possible name, especially rare ones. Third, the incoming name does not have any variant spelling of name(s) in the database (i.e. Western European last names). Fourth, there are virtually millions of potential name combinations or forms (i.e. East Asian first names).

The last three reasons are problems which prevent incoming personal names from being verified correctly by the database. Without finding an exact match in the name verification database, the personal name problem in this paper refers to scenario where ordering and gender (possibly culture, authenticity, and age group) cannot be determined correctly and automatically for every incoming personal name. Therefore, additional processing is required.

There are three different and broad application categories of related work in name matching (Borgman and Siegfried, 1992):

1. **Information retrieval:** Finding exact or variant form(s) of incoming name in verification database with no changes to the database. This present work is the most similar to ours where an incoming personal name is used as a search key to retrieve first/last name and gender information.
2. **Name authority control:** Mapping the incoming name upon initial entry to the most commonly used form in database. Current publications on author citation matching within the ACM (Association of Computing Machinery) portal database are examples of this (Feitelson, 2004). Unlike author names where the first names are usually abbreviated, many personal names in large databases have complete first and last names.
3. **Record linkage/duplication detection:** Detecting duplicates for incoming multiple data streams at input or during database cleanup. Recent publications focused on supervised learning on limited labelled data (Tejada *et al*, 2002) and on approximate string matching (Bilenko *et al*, 2003). Unlike their matching work which uses comparatively smaller data sets and has other informative address and phone data. Intelligently matching incoming names-only data with a comprehensive verification database seems like a harder problem.

Other specific applications of personal name matching include art history (Borgman and Siegfried, 1992), name entity extraction from free text (Cohen and Sarawagi, 2004; Patman and Thompson, 2003; Bikel *et al*, 1999), genealogy, law enforcement (Wang *et al*, 2004), law (Navarro *et al*, 2003; Branting, 2003), and registry identity resolution (Stanford ITS, 2005). Name matching has been explicitly or implicitly researched under databases, digital libraries, machine learning, natural language processing, statistics, and other research communities; and also known as identity uncertainty, identity matching, and name disambiguation.

MAIN FOCUS

In this section, the data sets, evaluation metrics, and computational resources are described. This is followed by an empirical comparison and discussion of results using four approaches for personal name analysis. The experimental Verification Data Set (VDS) has 99,570 unique names (examples) which are evenly distributed amongst gender, culture, and nationality. The original data set has 1 attribute (name component) and 3 class labels (first/last, gender, and culture). The Scoring Data Set (SDS) has 8,623 personal names/instances (each with first and last names) and is an assembly of four publicly available data sources which are multi-cultural and within the Australian context (all labelled with first/last and manually labelled with gender). To evaluate the effectiveness across all four approaches, this paper proposes a simple-to-use positive-valued Normalised Net Value (NNV) - to sum up all scores for each name component in the test set SDS.

Every name component is given a score s_i:

$$s_i = \begin{cases} 1 & \text{if correct match/prediction} \\ 0 & \text{if no match/prediction} \\ -1 & \text{if incorrect match/prediction} \end{cases}$$

Every approach on N number of names in the SDS is given an NNV:

$$\text{NNV} = \frac{\sum_{i=1}^{N} s_i + N}{2N}$$

Classifiers are also evaluated by F-measure (Witten and Frank, 2005):

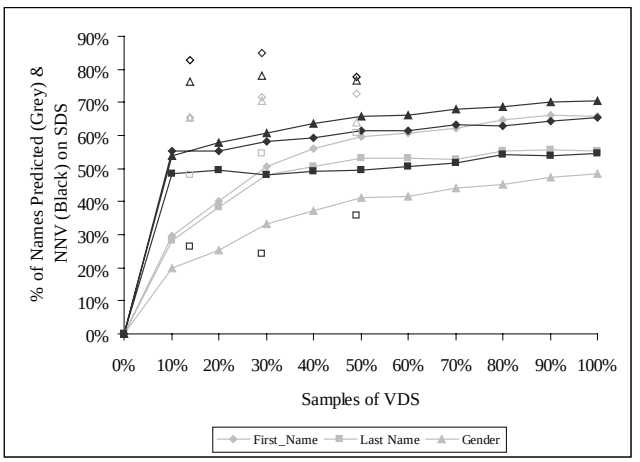
$$\text{F} - \text{measure} = \frac{2 \text{ x recall x precision}}{\text{recall } + \text{ precision}}$$

Exact-Matching Results

Each component in SDS is compared to every example in the VDS to find exact matches. For exact matching, phonetics, and simmetrics, when a match is found, its corresponding ordering and gender labels are counted, and the sum of candidate matches are returned. The label(s) with the highest count will be the final output. For example, if the name component "CLIFTON" is recognised in the VDS as both a first name and last name, the final prediction will be "F|L". But, it is important to note that in all subsequent experiments, predictions with conflicting outcomes such as "F|L" will be considered as a "no match/prediction" even though the correct class label is "F".

Figure 2 above shows the effect of having a larger random sample of VDS on the exact matching of first names, last names, and gender. Exact matching with the entire VDS can barely determine name authenticity, as it will involve manually checking 34.2% to 51.6%

Figure 2. Results of exact matching with different sample sizes and data selections of VDS on SDS



of the SDS which has no matches from the VDS. Its highest NNV is with the entire VDS: first names is 65.5%, of last names is 54.7%, and of gender is 70.6%. With larger data samples, the number of predictions (plotted lines in light grey) and normalised net value (plotted lines in dark grey) increase marginally. Interestingly, although the number of matches on gender is the lowest among the three, gender predictions are the most accurate.

In Figure 2, the hollow scatter points at 14%, 29%, and 49% are selected samples of the VDS which represent the choice of English (ENG) names only, Western European names only: ENG, French (FRE), German (GER), Dutch (DUT), Irish (IRI), Welsh (WEL); and Western European plus five other major name groups: Chinese (CHI), Indian (IND), Spanish (SPA), Arabic (ARA), Russian (RUS) accordingly. The dark grey hollow scatter points illustrate high accuracy for first name and gender probably because first names are repetitive and gender is determined from first names.

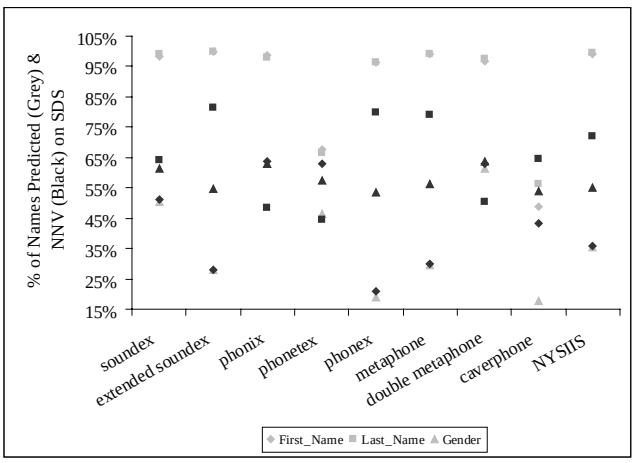
Phonetics Results

Phonetics (sounds-like algorithms) operate by encoding the SDS instance and then matching that against the encoded names from the VDS. The ones used in the experiments here are briefly described below:

1. **Soundex:** Soundex is the most widely used phonetic algorithm, has seven numerical codes and transforms the name into a four-letter code with the first letter of name preserved. Extended Soundex is a "relaxed" version of the original proposed by this paper: no preservation of the first letter of name, not restricted to a four-letter string, no trailing zeros to "pad" the four-letter code, and place "R" in the same numerical code group as "L".
2. **Soundex variants:** Phoni**x** applies more than a hundred string standardisation rules before a nine numerical code transformation (Zobel and Dart, 1996). Phonetex (Hodge and Austin, 2003) is a combination of Soundex and Phoni**x**. Phone*x* (Lait and Randell, 1993) is a combination of Soundex and Metaphone.
3. **Metaphone and extension:** Metaphone (Philips, 1990) reduces English words into sixteen consonant sounds and Double Metaphone (Philips, 2000) to twelve consonant sounds.
4. **Others:** Caverphone has more than fifty sequential rules and New York State Identification and Intelligence System (NYSIIS) uses five sets of rule sets to map name into code (Hood, 2002).

There are two obvious flaws in phonetics: Most of them were mainly designed for the Anglo-Saxon last names, yet many last names in most databases are of many different cultures. Also, many of them were designed for different purposes or were not general enough for widespread use. For example, Phonetex was designed for spell checkers; Phone*x* was adapted to British surnames, Caverphone was designed for New Zealand accents, and NYSIIS was mainly used for the New York city population.

Figure 3. Results of phonetics on first/last name and gender prediction on SDS



In Figure 3, there are phonetical matches to almost all SDS instances except for Phonetex and Caverphone. The highest NNV of first name is Phon**i**x at 63.9%, of last name is extended Soundex at 81.4%, and of gender is double Metaphone at 63.7%. The superior performance of extended Soundex against other more complex phonetical techniques for this problem is unexpected. The results confirm that phonetics were originally designed and, in comparison to simmetrics and classification, is still more effective for matching last names. In contrast, first names are neglected by phonetics.

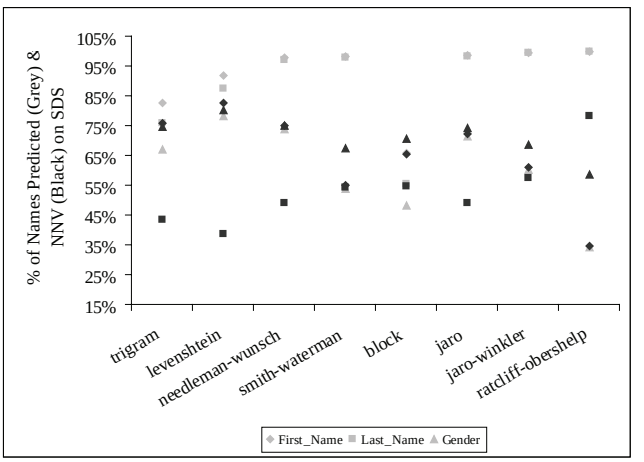
Simmetrics Results

While phonetics examine dissimilar groups of letters to identify similarities in pronunciation, simmetrics use similar or identical groups of letters. Simmetrics (looks-like algorithms, commonly known as edit distance algorithms) are character-based similarity metrics (Chapman, 2005; Cohen *et al*, 2003; Ratcliff and Metzener, 1998) which output the scaled similarity score (between 0 and 1) between two names components, where 0 represents totally different and 1 represents identical. Similarity measure = 1 - distance measure. Those used in the experiments here are briefly described below:

- **Edit distances:** Levenshtein calculates minimal number of single-character insertions, deletions, and substitutions to transform one string into another. Needleman-Wunsch is an extension of Levenshtein with a variable gap cost for insertions and deletions. Smith-Waterman is similar to Levenshtein but allows for context dependent gap costs.
- **Non-edit distances:** *N*-gram calculates the percentage of matches of all possible substrings of size n between two strings. Trigram is used in the experiments here. Block is the absolute difference of coordinates between two names. Jaro is dependent on number and order of common characters between two names. Jaro-Winkler, an extension of Jaro, takes in account the length of common prefix between two names. Ratcliff-Obershelp matches characters in the longest common subsequence, and recursively matching characters on the both sides of the longest common subsequence.

With similarity threshold set at 0.8, Figure 4 shows that simmetrics enables a very high match rate between VDS and SDS except for Trigram, Levenshtein, and Block. The highest NNV of first name is Levenshtein at 82.4%, of last name is Ratcliff-Obershelp at 78.1%, and of gender is Levenshtein at 80.3%. Like extended Soundex for last names, the basic Levenshtein produces the best NNV for first name and gender. It outperforms phonetics and classification because some first names are slight variants of the original. However, the percentage of names matched/predicted is relatively low at 91.7% and 78.2% respectively. Perhaps these names without matches/predictions should be manu-

Figure 4. Results of simmetrics (a group of string similarity metrics) on first/last name and gender prediction on SDS



ally investigated. There are other theoretically sound similarity metrics which have been experimented on VDS and SDS, but they are inefficient (Editex, Gotoh, and Monge-Elkan), or ineffective (Hirschberg and Ukkonen) for this problem.

Classification Results

Classifiers (discriminant functions) are trained with certain subsets of the VDS and score the SDS accordingly (it is the only approach which gives a ranked output). This is a hard problem because of the large numbers of nominal attribute values in the *n*-gram attributes. Due to this, typical decision trees (cannot handle identifier-like attributes) and support vector machines (extensive pre-processing required to convert data to numerical format) are not suited to this task. On the other hand, naïve Bayes is the only suitable classification algorithm which is extremely efficient and still comparable to other state-of-the-art algorithms.

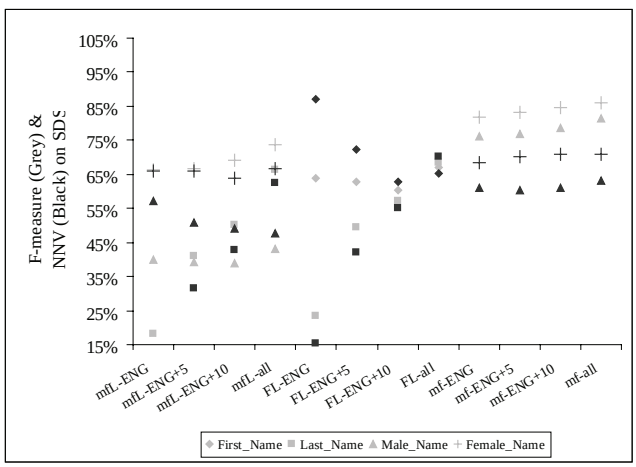
With reference to the x-axis of Figure 5 above, the classifiers are built from mfL which has a trinary class label (male, female, or last), FL which has a binary class label (first or last), and mf (from first names only) which has a binary class label (male or female). Finer-grained classifiers are built using selected samples of the VDS described before (ENG, ENG + 5, ENG + 10). The highest NNV of first name is FL-ENG at 87.0%, of last name is FL-all at 70.1%, of male name is mf-all at 63.0%, and of female name is mf-all at 71.0%. Although there is a conflict where the highest F-measure of first name is FL-all instead of FL-ENG, the latter is still preferred for SDS as most first names will be of English origin (even for different cultures). Also, it seems that results with binary class labels are slightly better than the trinary class label. The main disadvantage of classifiers, compared to the rest, is in the pre-processing effort required to transform data into a suitable format and the expertise needed to interpret the results.

Hybrid Results

Previous research claims that combination of results/evidence almost always improves performance (Hsiung, 2004; Hodge and Austin, 2003; Zobel and Dart, 1996), but this depends on the diversity of the individual results. First name and gender predictions are the best by combining results from simmetrics and classifiers where NNV is at 78.6% and 78.8% respectively. For last name, phonetics remain the single most important algorithm (combining results decreases NNV significantly here). The NNVs here are a few percentage points lower than the highest NNVs from other approaches but all the names from the SDS will now have a match/prediction to indicate whether a name component is a first or last (ordering), and/or male or female (gender) name.

There are three main lessons learnt from these experiments for personal name analysis. First, exact matching is inadequate. Second, more complex phonetical encoding and string-similarity algorithms often perform poorly in both accuracy and speed than the original ones. Therefore, there is no need to research more into this area. On the other hand, we also learnt that

Figure 5. Results of classification with different class labelling and data selection on SDS



the classification can play an essential part in personal name analysis, particularly when using hybridising it with simmetrics, to determine first name and gender.

FUTURE TRENDS

Personal name analysis is becoming more prevalent and important, especially in intelligence- and security-related research and applications:

1. **Research:** Author recognition in digital libraries, database marketing, fraud detection, homeland security, recommendation systems, and social network analysis.
2. **Private sector:** Account opening application; address change, payment activities in internet-based businesses; marketing in insurance, financial, and telecommunication businesses.
3. **Public sector:** Name verification activities in card issuing authorities, customs, electoral registers, health sectors, law enforcement, immigration, social security, and tax offices.

Opportunities for further research include more robust techniques to maintain individual privacy and confidentiality; use of other classification techniques, and experimentation with more test data sets; and application of the approaches discussed in this paper to extract culture, authenticity, and age group from personal names. As more personal name data with demographic information are become publicly available online (in the millions), this could well be a new data mining application area.

CONCLUSION

In this paper, personal name problem is defined for the case where the ordering and gender cannot be determined correctly and automatically for every incoming personal name. The recommended solution is to use the data from VDS with a set of techniques made up of phonetics (extended Soundex), simmetrics - string-similarity algorithms (Levenshtein), and classifiers (naïve Bayes algorithm). The combination of results/evidence from the above techniques improved performance, especially for predicting first names and gender. In addition, this paper emphasised the inadequacies of exact matching and scalability issues of more complex phonetical encoding and string-similarity algorithms.

REFERENCES

Bikel, D., Schwartz, R., & Weischedel, R. (1999). An Algorithm that Learns What's in a Name, *Machine Learning*, 34, 211-231.

Bilenko, M., Mooney, R., Cohen, W., Ravikumar, P., & Fienberg, S. (2003). Adaptive Name Matching in Information Integration, *IEEE Intelligent Systems*, 18(5), 16-23.

Borgman, C. & Siegfried, S. (1992). Getty's Synoname and Its Cousins: A Survey of Applications of Personal Name-Matching Algorithms, *Journal of the American Society for Information Science*, 43(7), 459-476,.

Branting, K. (2002). Name-matching Algorithms for Legal Case-management Systems, *Journal of Information, Law and Technology*.

Chapman S. (2005). SimMetrics – Open Source Similarity Measure Library. Retrieved April 2005, http://sourceforge.net/ projects/ simmetrics/.

Cohen, W., Ravikumar, P. & Fienberg, S. (2003). A Comparison of String Distance Metrics for Name Matching Tasks, *in Proceedings of AAAI03*.

Cohen, W. & Sarawagi, S. (2004). Exploiting Dictionaries in Named Entity Extraction: Combining Semi-Market Extraction Processes and Data Integration Methods, *In Proceedings of SIGKDD04*.

Feitelson, D. (2004). On Identifying Name Equivalences in Digital Libraries, *Information Research*, 9(4).

Hodge, V. & Austin, J. (2003). A Comparison of Standard Spell Checking Algorithms and a Novel Binary Neural Approach, *IEEE Transactions on Knowledge and Data Engineering*, 15(5), 1073-1081.

Hood, D. (2002). Caverphone: Phonetic Matching Algorithm, Technical Paper CTP060902, University of Otago, New Zealand.

Hsiung, P. (2004). Alias Detection in Link Data Sets, Technical Report CMU-RI-TR-04-22, Robotics Institute, Carnegie Mellon University, USA.

Lait, A. & Randell, B. (1993). An Assessment of Name Matching Algorithms, Technical Report, Department of Computing Science, University of Newcastle upon Tyne, UK.

Levitt, S. & Dubner, S. (2005). *Freakonomics: A Rogue Economist Explores the Hidden Side of Everything*, Allen Lane, Great Britain.

Navarro, G., Baeza-Yates, R. & Arcoverde, J. (2003). Matchsimile: A Flexible Approximate Matching Tool for Searching Proper Names, *Journal of the American Society for Information Science and Technology*, 54(1), 3-15.

Patman, F. & Thompson, P. (2003). Names: A New Frontier in Text Mining, *in Proceedings of Intelligence and Security Informatics*, 27-38.

Philips, L. (1990). Hanging on the Metaphone, *Computer Language*, 7(12).

Philips, L. (2000). The Double Metaphone Search Algorithm, *C/C++ Users Journal*.

Ratcliff, J. & Metzener, D. (1998). Ratcliff-Obershelp Pattern Recognition, *Dictionary of Algorithms and Data Structures*, Black P (ed.), NIST.

Stanford Information Technology Systems and Services. (2005). Person Registry Identity Resolution. Retrieved from April 2005, http://www.stanford.edu/dept/ itss/infrastructure/registry/roject/person_registry/attributes/matching.html.

Tejada, S., Knoblock, C. & Minton, S. (2002). Learning Domain-independent String Transformation Weights for High Accuracy Object Identification, *in Proceedings of SIGKDD02*.

Wang, G., Chen, H. & Atabakhsh, H. (2004). Automatically Detecting Deceptive Criminal Identities, *Communications of the ACM*, 47(3), 71-76.

Witten, I. & Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques with Java*, Morgan Kauffman Publishers, California, USA.

Zobel, J. & Dart, P. (1996). Phonetic String Matching: Lessons from Information Retrieval, *in Proceedings of 19th International Conference on Research and Development in Information Retrieval*, 166-172.

KEY TERMS

Attribute Construction: Manual or automatic creation of derived attributes from existing attributes to increase predictiveness or performance results.

Classification: A statistical procedure in which examples are placed into groups based on quantitative information on one or more characteristics inherent in the items examples, and based on a training set of previously labeled items.

Data Cleaning: The process of improving the quality of the data by modifying its form/content.

Fraud Detection: The application of mathematical algorithms to tease out evidences of fraud or abuse from available data where only the most suspicious ones are manually investigated.

Personal Name Data: Commercial or security-related databases which contain important personal identifiers such as personal names.

Security and Intelligence Informatics: The study of the development and use of advanced information technologies and systems for national and international security-related applications.

Similarity/Deception/Alias Detection: The search for the same entity which is masked or disguised as something else, either accidentally or deliberately.

Perspectives and Key Technologies of Semantic Web Search

Konstantinos Kotis
University of the Aegean, Greece

INTRODUCTION

Current keyword-based Web search engines (e.g. Google[a]) provide access to thousands of people for billions of indexed Web pages. Although the amount of irrelevant results returned due to polysemy (one word with several meanings) and synonymy (several words with one meaning) linguistic phenomena tends to be reduced (e.g. by narrowing the search using human-directed topic hierarchies as in Yahoo[b]), still the uncontrolled publication of Web pages requires an alternative to the way Web information is authored and retrieved today. This alternative can be the technologies of the new era of the Semantic Web.

The Semantic Web, currently using OWL language to describe content, is an extension and an alternative at the same time to the traditional Web. A Semantic Web Document (SWD) describes its content with semantics, i.e. domain-specific tags related to a specific conceptualization of a domain, adding meaning to the document's (annotated) content. Ontologies play a key role to providing such description since they provide a standard way for explicit and formal conceptualizations of domains. Since traditional Web search engines cannot easily take advantage of documents' semantics, e.g. they cannot find documents that describe similar concepts and not just similar words, semantic search engines (e.g. SWOOGLE[c], OntoSearch[d]) and several other semantic search technologies have been proposed (e.g. Semantic Portals (Zhang et al, 2005), Semantic Wikis (Völkel *et al*, 2006), multi-agent P2P ontology-based semantic routing (of queries) systems (Tamma *et al*, 2004), and ontology mapping-based query/answering systems (Lopez *et al*, 2006; Kotis & Vouros, 2006, Bouquet *et al*, 2004). Within these technologies, queries can be placed as formally described (or annotated) content, and a semantic matching algorithm can provide the exact matching with SWDs that their semantics match the semantics of the query.

Although the Semantic Web technology contributes much in the retrieval of Web information, there are some open issues to be tackled. First of all, unstructured (traditional Web) documents must be semantically annotated with domain-specific tags (ontology-based annotation) in order to be utilized by semantic search technologies. This is not an easy task, and requires specific domain ontologies to be developed that will provide such semantics (tags). A fully automatic annotation process is still an open issue. On the other hand, SWDs can be semantically retrieved only by formal queries. The construction of a formal query is also a difficult and time-consuming task since a formal language must be learned. Techniques towards automating the transformation of a natural language query to a formal (structured) one are currently investigated. Nevertheless, more sophisticated technologies such as the mapping of several schemes to a formal query constructed in the form of an ontology must be investigated. The technology is proposed for retrieving heterogeneous and distributed SWDs, since their structure cannot be known a priory (in open environments like the Semantic Web).

This article aims to provide an insight on current technologies used in Semantic Web search, focusing on two issues: a) the automatic construction of a formal query (*query ontology*) and b) the querying of a collection of knowledge sources whose structure is not known a priory (distributed and semantically heterogeneous documents).

BACKGROUND

A keyword-based Web search mainly concerns search techniques that are based on string (lexical) matching of the query terms to the terms contained in Web documents. Traditionally, keyword-based search is used for unstructured Web documents' (text with no semantics

attached) retrieval, where retrieval is obtained when query terms are matched to terms found in documents. Several techniques for keyword-based Web search have been introduced (Alesso, 2004), with the most popular being the simple Boolean search, i.e. combination of keywords based on Boolean operators AND, OR, NOT. Other techniques include

- wildcard and proximity search (syntactic analysis of documents or query terms),
- fuzzy search (handles misspelling and plural variations of keywords),
- contextual search (analyse the content of Web pages and return the subject of the page),
- keyword location-based search (keywords occurring in the title tags of the Web page are more important than those in the body),
- human(or topic)-directed search (use of topic hierarchies, manually created, to help users to narrow the search and make search results more relevant),
- thesaurus-based search (use specific relations such as synonym to help retrieve relevant information even if keyword is not present in a document),
- and finally statistics-based search such as Google's PageRank[e] technology.

Keyword-based search technology has been also used to retrieve SWDs by matching NL query terms to terms that lexicalize concepts of a SWD (e.g. an ontology concept). Such technology, when used in semantic search engines (e.g. SWOOGLE), do not utilize the semantics of the SWD in the matching algorithm. Matching is based on lexical techniques (string matching of keywords with terms that lexicalize concepts of a an ontology) although the retrieved content is semantically described (i.e. SWDs). Generally, semantic matching is performed in extension to the lexical one and the syntactic similarity between terms is not of interest. In fact, what is important is the similarity of the meaning of two terms. For instance, a match between a query-term "book" and a document-term "reserve" may be correctly identified if the sense of concept "book" is "the reservation of a ticket" (synonymy). On the other hand, a match between the term "book" found in a query and an identical term found in a Web document, may be incorrectly identified if their senses are completely different i.e. the query-term "book", meaning a publication, and the document-term "book", meaning a reservation (polysemy).

Semantic matching requires that the semantics of both the query and the document must be known or uncovered prior their matching. If the query is formally specified, the semantics of each term can be explicitly defined. Thus, if a query is represented as an ontology (query ontology), the semantics of each term that lexicalizes an ontology concept can be revealed by the semantic relations between this concept and the other concepts of the ontology (structure of its neighborhood). Such semantic relations are not only subsumption (is-a) relations, but also others such as "part-of", "meronym", "synonym", etc. On the other hand, if the query is informally specified, i.e. in natural language, the semantics of each term in the query must be somehow uncovered. The issue here is how a machine can "guess" what the intended meaning of an informal query is, in order to retrieve the document that is closer to this meaning and therefore, more interesting to the user. Intelligent search engines such as AskJeeves[f] (Teoma technology) try to tackle this issue by analysing the terms and their relations in a sophisticated way using natural language processing techniques or by refining the query in collaboration with the users. An alternative technique map each term of a query to its intended meaning (sense found in lexicon), using a combination of vector space indexing techniques such as LSI (Deerwester et al, 1990) and a lexicon such as WordNet (Miler, 1995). Furthermore, to be able to execute a semantic matching, the document (in addition to the query) must also provide its semantics. In case of a SWD, the semantics of the document are formally and explicitly specified in an ontology. In case of unstructured documents, advanced ontology learning techniques are required in order to extract their semantics and use them to annotate the related documents.

Further related work has been also carried out and presented (Karanastasi & Christodoulakis, 2007), where an ontology-driven semantic ranking methodology for ontology concepts is used for natural language disambiguation. This work has been proposed in the context of OntoNL framework (Karanastasi et al, 2007). The methodology uses domain specific ontologies for the semantic disambiguation. The disambiguation procedure is automatic and quite promising.

There are several other proposals concerning the retrieval of SWDs. The majority of them assume that

the query is given in a structured way - using a formal language - and provide no advanced means for the (semantic) alignment of the query to the contents of distributed and heterogeneous SWDs. Querying such documents using a controlled language instead of a formal one to formulate a semantic query (Bernstein, 2005) still requires that users should learn a new language. Querying Semantic Web documents using formal queries requires that either the structure of SWDs is known a priory, (i.e. semantic homogeneity), and so use Semantic Web query languages such as OWL-QL (Fikes et al, 2003) and RQL (Karvounarakis, 2003) to query semantic portals, or the structure of SWDs is unknown (semantic heterogeneity) and so the querying is performed either by using a global schema (shared common ontology) to map queries on it or (in case of distributed settings i.e. p2p approaches) performing horizontal mappings across local schemas.

MAIN FOCUS

Latest research work (Lopez *et al*, 2006a; Kotis & Vouros, 2006) builds towards a method for a) automatically approximating the meaning of a simple NL query, reformulating it into an ontology (*query ontology*) that expresses the intended meaning of the query terms and of the query string as a whole, b) retrieving the most "relative" SWDs based on their similarity with the query ontology. The setting of such approaches is addressed to users with no background knowledge in formal languages and to SWDs that are heterogenous (capture different domain knowledge) and distributed (no centralized knowledge about their content exists).

An innovative technology to the transformation of an NL query to a query ontology is provided by the mapping of each query term to a sense in a semantic lexicon (e.g. WordNet) and by consulting the semantic relations between senses. Furthermore, the use of query ontologies to retrieve distributed and heterogeneous SWDs points to the need of measuring the similarity between the content of the query and that of SWDs. This is done by producing and comparing ontology mappings.

NL Query Reformulation for Ontology Construction

The mapping of terms to WordNet senses is performed by computing a semantic morphism that can be considered as a similarity function between terms and WordNet senses (Kotis *et al,* 2006). For the computation of this similarity the s-morphism takes into account the vicinity of each ontology term. Since this computation is based on the hypothesis that query terms are related to each other, the vicinity of a query term includes all the other terms in the query.

WordNet lexicon contains lexical and semantic information about nouns, verbs, adverbs, and adjectives, organized around the notion of a synset. A synset is a set of words with the same part-of-speech that can be interchanged in a certain context. A synset is often further described by a gloss. Semantic relations among synsets include among others the synonymy, hyper(hyp)onymy, meronymy and antonymy relations. WordNet contains thousands of words, synsets and links between concepts. Any semantic lexicon can be used for building a query ontology: For instance, prominent and well-agreed ontologies or thesauri that are known to the users of a retrieval service may be exploited for this purpose. However, this is only a conjecture that remains to be tested.

The s-morphism is computed by the Latent Semantic Indexing (LSI) method. LSI is a vector space technique originally proposed for information retrieval and indexing. It assumes that there is an underlying latent semantic space that estimates by means of statistical techniques using an association matrix (n×m) of term-document data (terms×WordNet senses in this case).

Each term in the initial NL query is being mapped to a WordNet sense through the computation of the s-morphism. Given the vicinity of the term (i.e. the other terms in the query) this sense is considered to express the meaning of the term, i.e. it reflects the intended meaning for the particular term in the context of the query. Given the computed mapping, the vicinity of each query term is being enriched by the set of the most important terms in the corresponding WordNet sense.

The reformulated enriched query can be used to retrieve unstructured documents using keyword-based search techniques, with higher precision than existing methods (Kotis, 2005). The length of the reformulated

query is rather important for the precision of the retrieval. However, to be able to retrieve SWDs, the query must be further processed towards the construction of the query ontology. The algorithm for this process is presented in (Kotis & Vouros, 2006).

Matching Queries Through Ontology Mapping

A variety of methods and tools have been proposed for solving the problem of ontology mapping, and although there is still much to be done, remarkable achievements have been made (Euzenat *et al*, 2006).

For the effective retrieval of SWDs an algorithm that computes the similarity of SWDs and the already constructed query ontology has been proposed. This similarity is being computed using an automatic ontology mapping tool called AUTOMS[g] that extends the HCONE-merge method for ontology mapping (Kotis et al, 2006) by combining lexical, semantic, and structural matching methods. The ranking of retrieved SWDs is computed based on how well they match to the query ontology: This is determined by the number of mappings (mapped concepts) between the query ontology and a SWD (Kotis and Vouros, 2006).

Lexical matching computes the matching of ontology concept names (labels at nodes), estimating the similarity among concepts using syntactic similarity measures. Structural matching computes the matching of ontology concepts by taking into account the similarity of concepts in their neighborhoods. The neighborhood of a concept includes those concepts that are related to it. Finally, semantic matching concerns the matching between the meanings of concept specifications. The computation of semantic matching may rely to external information found in lexicons, thesauri or reference ontologies, incorporating semantic knowledge (mostly domain-dependent) into the process.

AUTOMS tool, although in the experimental stage, achieves high precision and recall in mapping ontologies. Some first experiments and preliminary results of AUTOMS performance within a querying/answering SWDs approach can be found in (Kotis & Vouros, 2006).

Related Technologies

AquaLog (Lopez et al, 2005) is a fully implemented ontology-driven querying/answering system, which takes an ontology and a NL query as an input and returns answers drawn from semantic markup compliant with the input ontology. It is based on GATE infrastructure to obtain a set of syntactic annotations associated with the input query, extended by the use of JAPE grammars to identify terms, relations, question indicators, features, and to classify the query into a category. Knowing the category of the query and having the GATE annotations for the query, the system automatically creates query-triples. These query-triples are further processed and, using the structure and vocabulary of the input ontology, are mapped to ontology-compliant semantic markup or triples. An important limitation is that the system can be used to map a query only to one (shared organizational) ontology.

AquaLog is used in PowerAqua (Lopez et al 2006a), a novel querying/answering system which provides answers drawn from multiple, heterogeneous and distributed ontologies on the Web. Although promising technology, it is still in the phase of implementation, and currently no experiments have been given out any evaluation results. The core of this system is PowerMap (Lopez et al, 2006b) mapping algorithm which offers an alternative to current ontology mapping techniques by focusing on the real-time on-the-fly time-performance mapping of ontologies.

FUTURE TRENDS

We visualize querying/answering systems performing in a new Web, a unified Web of Knowledge (Kotis, 2005). Knowledge recorded in such a Web shall be *distributed* (querying more than one peer-source of knowledge), *heterogeneous* (querying knowledge with unknown structure), *personalized* (background knowledge extracted from already executed queries, predefined preferences on queries), *dynamic* (continuously evolving), and *open* (reached by anyone and from everywhere), providing a unified view of both structured and unstructured documents against simple natural language queries. New technologies should allow querying structured and unstructured documents in a unified way using keyword-based queries, as well as the querying of SWDs documents using both keywords and ontology queries.

Towards this vision, several issues must be further explored such as a) the impact of querying/answering history and/or the use of alternative (or combination

of) lexicons/thesauruses to the disambiguation of new queries, b) the degree of user involvement during the construction of query ontologies and/or the mapping of query ontologies to domain ones, and c) strategies for the maximization of performance (time, precision, recall) of ontology mapping algorithms.

CONCLUSIONS

The need to retrieve distributed and heterogeneous knowledge in open and dynamic environments such as the Semantic Web imposes the realization of new perspectives and the proposal of new approaches. These approaches accentuate the role of ontology mapping within Semantic Web search technology. Users must be empowered with new technologies to search the whole Semantic Web, technologies that will not require advanced skills, and at the same time that will ensure that all the query-related knowledge (and only that) will be promptly retrieved. Towards these targets, we have presented a review on latest related approaches, pointing also out the importance of such approaches to the success of Semantic Web.

REFERENCES

Alesso, P. (2004). Semantic Search Technology. AIS SIGSEMIS Bulletin 1(3), 86-98

Bernstein, A., Kaufmann, E., & Fuchs, N. (2005). Talking to the Semantic Web - A Controlled English Query Interface for Ontologies. *AIS SIGSEMIS Bulletin,* 2(1) pp. 42-47

Bouquet, P., Kuper, G. M., Scoz, M., & Zanobini, S. (2004). Asking and Answering Semantic Queries. Workshop on Meaning Coordination and Negotiation Workshop (MCN-04) in conjunction with the 3rd International Semantic Web Conference (ISWC-04), Hiroshima Japan

Deerwester, S., Dumais, S., T., Furnas, G., W., Landauer, T., K., & Harshman, R. (1990) Indexing by Latent Semantic Analysis. Journal of the American Society of Information Science. 41(6), 391-407

Euzenat, J., Mochol, M., Shvaiko, P., Stuckenschmidt, H., Šváb, O., Svátek, V., Hage, W., Yatskevich, M. (2006). Results of the Ontology Alignment Evaluation Initiative 2006. Proceedings of the 1st International Workshop on Ontology Matching (OM-2006), International Semantic Web Conference (ISWC-2006) Athens, Georgia, USA

Fikes, R., Hayes, P., & Horrocks, I. (2003). OWL-QL - a language for deductive query answering on the semantic web. Technical Report, Knowledge Systems Laboratory, Stanford, CA

Karanastasi A., Christodoulakis S. (2007). "Ontology-Driven Semantic Ranking for Natural Language Disambiguation in the OntoNL Framework", in the Proceedings of the 4th European Semantic Web Conference (ESWC), 3-7 June 2007, Innsbruck, Austria

Karanastasi A., Zwtos A., Christodoulakis S. (2007). "The OntoNL Framework for Natural Language Interface Generation and a Domain-Specific Application", in the Proceedings of the DELOS Conference on Digital Libraries 2007, 13-14 February 2007, Tirrenia, Pisa, Italy

Karvounarakis, G. (2003). The RDF query language (RQL). Technical report, Institute of Computer Science, Foundation of Research Technology

Kotis K. (2005). Using simple ontologies to build personal Webs of knowledge. 25th International Conference of the British Computer Society, SGAI, Cambridge, UK

Kotis, K. & Vouros, G. (2006). Towards Semantic Web Documents Retrieval through Ontology Mapping: Preliminary Results. 1st Asian Semantic Web Conference (ASWC 2006) Workshop on Web Search Technology - from Search to Semantic Search, Beijing, China

Kotis, K., Vouros, G., & Stergiou, K. (2006). Towards Automatic Merging of Domain Ontologies: The HCONE-merge approach. Elsevier's Journal of Web Semantics (JWS). 4(1), pp. 60-79.

Lopez, V., Motta, E., & Pasin, M. (2005). AquaLog: An Ontology portable Question Answering Interface for the Semantic Web. European Semantic Web Conference 2005, Heraklion, Crete.

Lopez, V., Motta, E., & Uren, V. (2006). PowerAqua: Fishing the Semantic Web. European Semantic Web Conference 2006, Montenegro.

Lopez, V., Sabou, M., & Motta, E. (2006). PowerMap: Mapping the Real Semantic Web on the Fly. International Semantic Web Conference., Georgia, Atlanta.

Miller, G. (1995). WordNet: A lexical database for English. Communications of the ACM, 38(11) pp. 39-41

Tamma, V., Blacoe, I., Lithgow-Smith, B., & Wooldridge M. (2004). SERSE: Searching for Semantic Web Content In *Proceedings of the Sixteenth European Conference on Artificial Intelligence (ECAI-04)*, Valencia, Spain

Völkel, M., Krötzsch, M., Vrandecic, D., Haller, H., Studer, R. (2006). Semantic Wikipedia**.** WWW 2006, Edinburgh, Scotland, May 23-26

Zhang, L., Yu, Y., & Yang, Y. (2005). An enhanced Model for Searching in Semantic Portals. WWW 2005, May 10-14, Chiba, Japan

KEY TERMS

Keyword-Based Web Search: Searching for Web documents that their content can be lexically matched (string similarity) with the content of a NL query.

Ontology Mapping: The mapping between two ontologies can be defined as a morphism from one ontology to the other i.e. a collection of functions assigning the symbols used in one vocabulary to the symbols of the other.

Ontology Matching: The computation of similarity functions in order to discover similarities between ontology concepts or/and properties pairs using combinations of lexical, structural, and semantic knowledge.

Query Ontology: An ontology automatically created by the mapping of NL query terms to a lexicon. The aim is to check its similarity against SWDs using ontology mapping techniques.

Semantic Knowledge: Knowledge that is captured by uncovering the human intended meaning of concepts, relying either on the computation of similarity functions that "translate" semantic relations (hyperonym, hyponym, meronym, part-of, etc) between the intension (the attribute set) of concepts or on the use of external information such as (the mappings of ontology concepts to) terms' meanings found in lexicons or thesauruses.

Semantic Search: Searching for SWDs with content that can be semantically matched (semantic similarity) with the content of a formal query

Semantic Web Document: Is a document represented as an RDF graph, written in RDF/XML syntax, allowing its content to be processable from machines. An ontology can be a special type of SWD (written in OWL syntax) that extends the semantics of RDF(S) language.

ENDNOTES

[a] http://www.google.com
[b] http://www.yahoo.com
[c] http://swoogle.umbc.edu/
[d] http://www.ontosearch.org/
[e] http://www.google.com/technology/
[f] http://www.ask.com/
[g] http://www.icsd.aegean.gr/ai%2Dlab/projects/AUTOMS/

A Philosophical Perspective on Knowledge Creation

Nilmini Wickramasinghe
Stuart School of Business, Illinois Institute of Technology, USA

Rajeev K. Bali
Coventry University, UK

INTRODUCTION

Today's economy is increasingly based on knowledge and information (Davenport, & Grover 2001). Knowledge is now recognized as the driver of productivity and economic growth, leading to a new focus on the role of information, technology and learning in economic performance. Organizations trying to survive and prosper in such an economy are turning their focus to strategies, processes tools and technologies that can facilitate them to create knowledge. A vital and well respected technique in knowledge creation is data mining which enables critical knowledge to be gained from the analysis of large amounts of data and information. Traditional data mining and the KDD process (knowledge discovery in data bases) tends to view the knowledge product as a homogeneous product. Knowledge however, is a multifaceted construct, drawing upon various philosophical perspectives including Lockean/Leibnitzian and Hegelian/Kantian, exhibiting subjective and objective aspects as well as having tacit and explicit forms (Nonaka, 1994; Alavi & Leidner, 2001; Schultze & Leidner, 2002; Wickramasinghe et al, 2003). It is the thesis of this discussion that by taking a broader perspective of the resultant knowledge product from the KDD process; namely by incorporating both a people-based perspective and process-based perspective into the traditional KDD process, it will not only provide a more complete and macro perspective on knowledge creation but also a more balanced approach to knowledge creation which will in turn serve to enhance the extant knowledge base of an organization and facilitate the realization of superior decision making. The implications for data mining are clearly far reaching and are certain to help organizations more effectively realize the full potential of their knowledge assets, improve the likelihood of using/re-using the created knowledge and thereby enable them to be well positioned in today's knowledge economy.

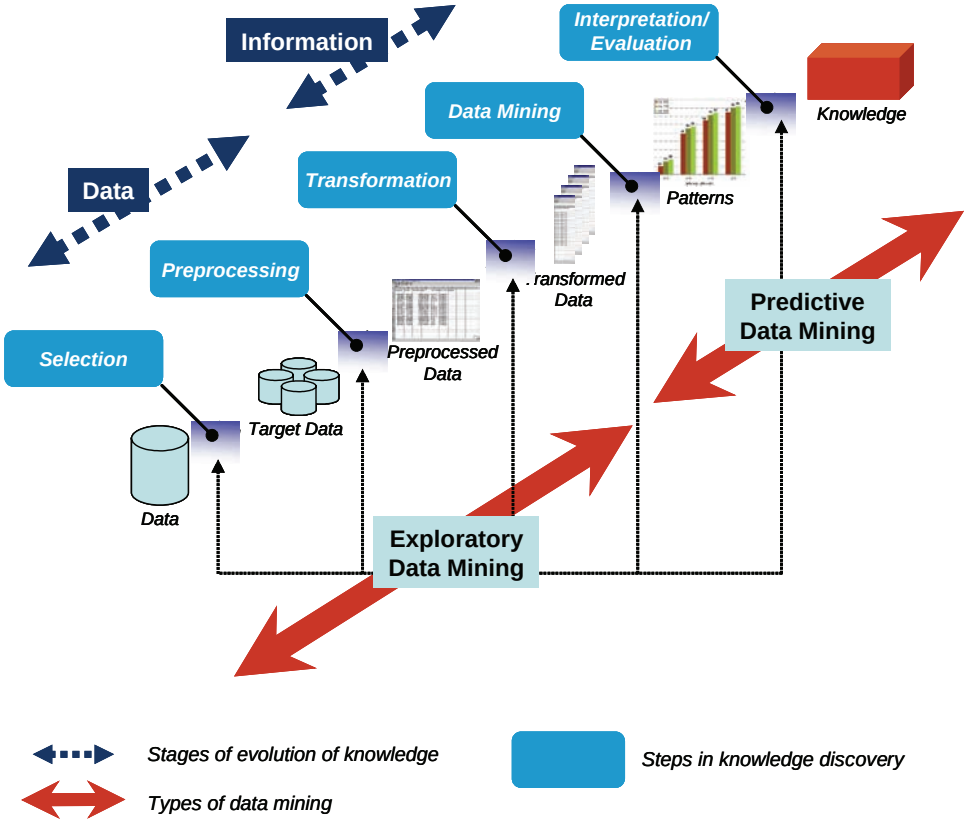
BACKGROUND: KNOWLEDGE CREATION

Knowledge Creation through Data Mining and the KDD Process

Knowledge discovery in databases (KDD), (and more specifically data mining) approaches knowledge creation from a primarily technology driven perspective. In particular, the KDD process focuses on how data is transformed into knowledge by identifying valid, novel, potentially useful, and ultimately understandable patterns in data (Spiegler, 2003; Fayyad et al., 1996). KDD is primarily used on data sets for creating knowledge through model building, or by finding patterns and relationships in data.

From an application perspective, data mining and KDD are often used interchangeably. Figure 1 presents a generic representation of a typical knowledge discovery process. This figure not only depicts each stage within the KDD process but also highlights the evolution of knowledge from data through information in this process as well as the two major types of data mining; namely, exploratory and predictive. whereas the last two steps (i.e., data mining and interpretation/evaluation) in the KDD process are considered predictive data mining. It is important to note in figure 1 that typically in the KDD process the knowledge component itself is treated as a homogeneous block. Given the well established multifaceted nature of the knowledge construct (Boland & Tenkasi, 1995; Malhotra, 2000; Alavi & Leidner, 2001; Schultze & Leidner, 2002; Wickramasinghe et al.,

Figure 1. Integrated view of the Knowledge Discovery Process (Adapted from Wickramasinghe et al, 2003)



2003) this would appear to be a significant limitation or over simplification of knowledge creation through data mining as a technique and the KDD process in general.

The Psycho-Social Driven Perspective to Knowledge Creation

Knowledge can exist as an object, in essentially two forms; explicit or factual knowledge and tacit or experiential i.e.," know how" (Polyani, 1958; 1966). Of equal significance is the fact that organizational knowledge is not static; rather it changes and evolves during the life time of an organization (Becerra-Fernandez, 2001; Bendoly, 2003; Choi & Lee, 2003). Furthermore, it is possible to change the form of knowledge; i.e., transform existing tacit knowledge into new explicit knowledge and existing explicit knowledge into new tacit knowledge or to transform the subjective form of knowledge into the objective form of knowledge (Nonaka & Nishiguchi, 2001; Nonaka, 1994). This process of transforming the form of knowledge, and thus increasing the extant knowledge base as well as the amount and utilization of the knowledge within the organization, is known as the knowledge spiral (Nonaka & Nishiguchi, 2001). In each of these instances the overall extant knowledge base of the organization grows to a new, superior knowledge base.

According to Nonaka (Nonaka & Nishiguchi, 2001): 1) Tacit to tacit knowledge transformation usually occurs through apprenticeship type relations where the teacher or master passes on the skill to the apprentice. 2) Explicit to explicit knowledge transformation usually occurs via formal learning of facts. 3) Tacit to explicit knowledge transformation usually occurs when there is an articulation of nuances; for example, as in healthcare if a renowned surgeon is questioned as to why he does a particular procedure in a certain manner, by his articulation of the steps the tacit knowledge becomes explicit and 4) Explicit to tacit knowledge transformation usually occurs as new explicit knowledge is internalized it can then be used to broaden, reframe and extend one's tacit knowledge. These transformations are often referred to as the modes of socializa-

tion, combination, externalization and internalization respectively (Nonaka, 1994). Integral to this changing of knowledge through the knowledge spiral is that new knowledge is created (Nonaka & Nishiguchi, 2001) and this can bring many benefits to organizations. Specifically, in today's knowledge-centric economy, processes that effect a positive change to the existing knowledge base of the organization and facilitate better use of the organization's intellectual capital, as the knowledge spiral does, are of paramount importance.

Two other primarily people driven frameworks that focus on knowledge creation as a central theme are Spender's and Blackler's respective frameworks (Newell et al, 2002; Swan et al, 1999). Spender draws a distinction between individual knowledge and social knowledge, each of which he claims can be implicit or explicit (Newell et al, 2002). From this framework we can see that Spender's definition of implicit knowledge corresponds to Nonaka's tacit knowledge. However, unlike Spender, Nonaka doesn't differentiate between individual and social dimensions of knowledge; rather he just focuses on the nature and types of the knowledge itself. In contrast, Blackler (Newell et al., 2002) views knowledge creation from an organizational perspective, noting that knowledge can exist as encoded, embedded, embodied, encultured and/or embrained. In addition, Blackler emphasized that for different organizational types, different types of knowledge predominate and highlighted the connection between knowledge and organizational processes (Newell et al., 2002).

Blackler's types of knowledge can be thought of in terms of spanning a continuum of tacit (implicit) through to explicit with embrained being predominantly tacit (implicit) and encoded being predominantly explicit while embedded, embodied and encultured types of knowledge exhibit varying degrees of a tacit (implicit) /explicit combination.

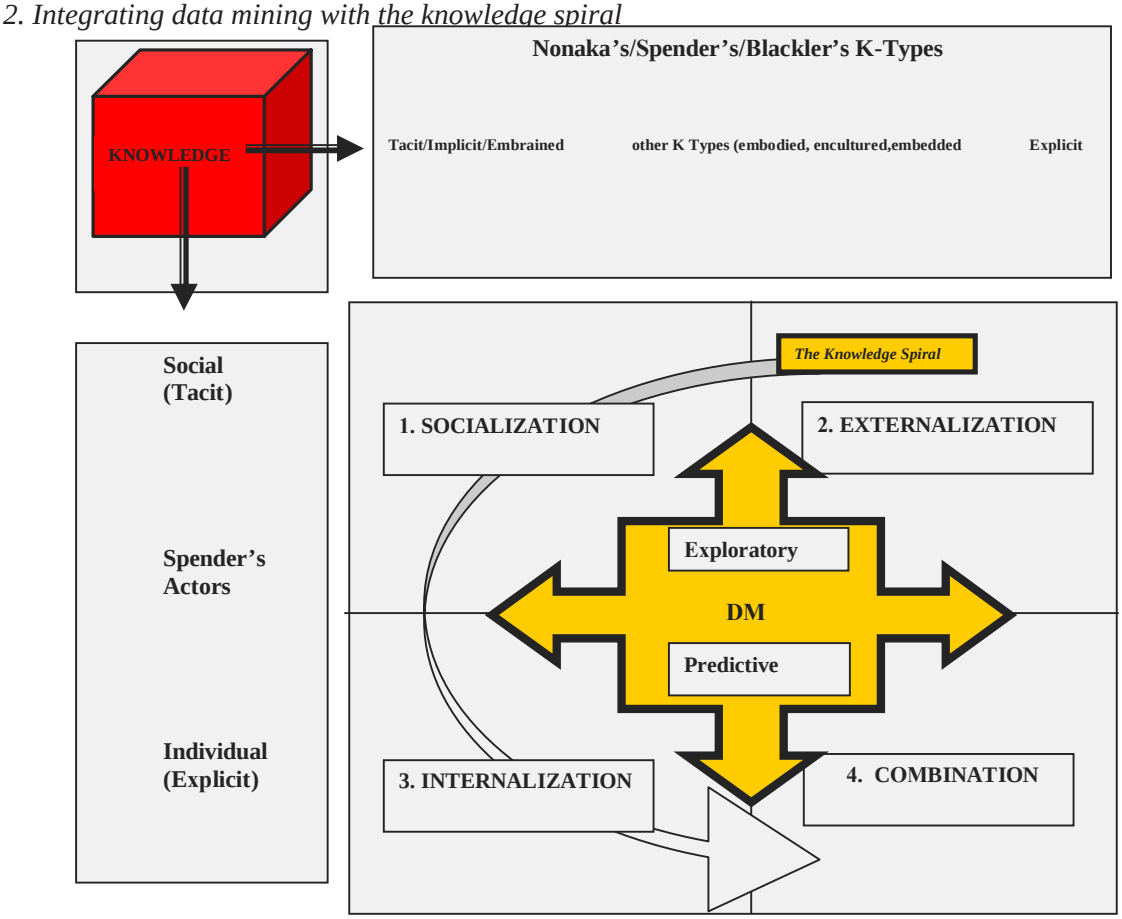
MAIN THRUST: ENRICHING DATA MINING WITH THE KNOWLEDGE SPIRAL

To conceive of knowledge as a collection of information seems to rob the concept of all of its life... Knowledge resides in the user and not in the collection. It is how the user reacts to a collection of information that matters. Churchman (1971, p. 10).

Churchman is clearly underscoring the importance of people in the process of knowledge creation. However, most formulations of information technology (IT) enabled knowledge management, and data mining in particular seem to have not only ignored the ignored the human element but also take a very myopic and homogenous perspective on the knowledge construct itself. Recent research that has surveyed the literature on KM indicates the need for more frameworks for knowledge management and particularly a meta framework to facilitate more successful realization of the KM steps (Wickramasinghe & Mills, 2001; Holsapple & Joshi, 2002; Alavi & Leidner, 2001; Schultze & Leidner, 2002). From a macro knowledge management perspective, the knowledge spiral is the cornerstone of knowledge creation. From a micro data mining perspective one of the key strengths of data mining as a technique is that it facilitates knowledge creation from data. Therefore, by integrating the algorithmic approach of knowledge creation (in particular data mining) with the psycho-social approach of knowledge creation (i.e., the people driven frameworks of knowledge creation, in particular the knowledge spiral) it is indeed possible to develop a meta framework for knowledge creation. By so doing, a richer and more complete approach to knowledge creation is realized. Such an approach not only leads to a deeper understanding of the knowledge creation process but also offers a knowledge creation methodology that is more customizable to specific organizational contexts, structures and cultures. Furthermore, it brings the human factor back into the knowledge creation process and doesn't over simplify the complex knowledge construct as a homogenous product.

Specifically, in figure 2 the knowledge product of data mining is broken into its constituent components based on the people driven perspectives (i.e., Blackler, Spender and Nonaka, respectively) of knowledge creation. On the other hand, the specific modes of transformation of the knowledge spiral discussed by Nonaka in his Knowledge Spiral should benefit from the algorithmic structured nature of both exploratory and predictive data mining techniques. For example, if we consider socialization which is described in (Nonaka & Nishiguchi, 2001, Nonaka, 1994) as the process of creating new tacit knowledge through discussion within groups, more specifically groups of experts, and we then incorporate the results of data mining techniques into this context this provides a structured forum and

Figure 2. Integrating data mining with the knowledge spiral



Specific Modes of Transformations and the Role of Data Mining in Realizing Them:

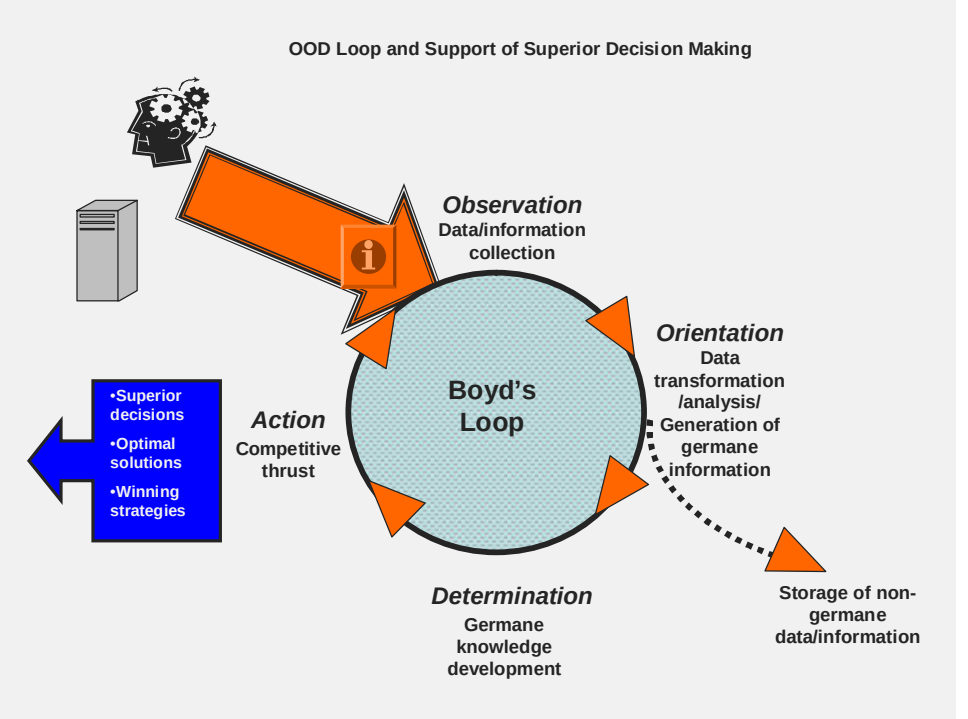
1. Socialization – experiential to experiential via practice group interactions on data mining results
2. Externalization – written extensions to data mining results
3. Internalization – enhancing experiential skills by learning from data mining results
4. Combination – gaining insights through applying various data mining visualization techniques

hence a "jump start" for guiding the dialogue and consequently knowledge creation. We note here though that this only enriches the socialization process without restricting the actual brainstorming activities, and thus not necessarily leading to the side effect of truncating divergent thoughts. This also holds for Nonaka's other modes of knowledge transformation.

Hierarchically, gathering of information precedes transformation of information into useable knowledge (Massey et al., 2002; Alavi & Leidner, 2001). Hence, the rate of information collection and the quality of the collected information will have a major impact on the quality (usefulness) of the generated knowledge (Chang et al., 2005). In dynamic and unstable environments, relative to the environment, the decision maker is in a position of tremendous information inferiority. In order to make effective decisions he/she must rapidly process seemingly irrelevant data and information into relevant and useable knowledge(Courtney, 2001; Drucker, 1993; Boyd, 1976; Award & Ghaziri, 2004; Newell et al., 2002, Schultz and Leidner, 2002; Wickramasinghe, 2006). This necessitates a process perspective to knowledge management (Wickramasinghe and von Lubitz, 2006; von Lubitz and Wickramasinghe, 2005b). The cornerstone of such a perspective is the OODA Loop (Figure 3) which provides formalized analysis of the processes involved in the development of a superior strategy (Boyd, 1976; 1987; von Lubitz & Wickramasinghe, 2006).

The Loop is based on a cycle of four interrelated stages revolving in time and space: Observation followed by Orientation, then by Decision, and finally

Figure 3. (Adapted from von Lubitz and Wickramasinghe, 2005ab)



Action. At the Observation and Orientation stages, multispectral implicit and explicit inputs are gathered (Observation) and converted into coherent information (Orientation). The latter determines the sequential Determination (knowledge generation) and Action (practical implementation of knowledge) steps. The outcome of the latter affects, in turn, the character of the starting point (Observation) of the next revolution in the forward progression of the rolling loop. The Orientation stage specifies the characteristics and the nature of the "center of thrust" at which the effort is to concentrate during the Determination and Action stages. Hence, the Loop implicitly incorporates the rule of "economy of force," i.e., the requirement that only minimum but adequate (containment) effort is applied to insignificant aspects of competitive interaction. The Loop exists as a network of simultaneous and intertwined events that characterize the multidimensional action space (competition space), and both influence and are influenced by the actor (e.g., an organization) at the centre of the network.

It is the incorporation of the dynamic aspect of the "action space" that makes the Loop particularly useful to environments that are inherently unstable and unpredictable i.e., medicine, business, war and emergency and disaster scenarios (von Lubitz and Wickramasinghe, 2005a; 2005b; 2006).

Thus as organizations strive to survive and thrive in today's competitive business environment incorporating people and process perspectives into their data mining initiatives will, through the extraction of germane knowledge, pertinent information and relevant data, serve to enable and support effective and superior decision making; a critical competitive necessity.

FUTURE TRENDS

The two significant ways to create knowledge are 1) synthesis of new knowledge through socialization with experts a primarily people dominated perspective 2) discovery by finding interesting patterns through observation and combination of explicit data, a primarily technology driven perspective (Becerra-Fernandez et al., 2004). In today's knowledge economy, knowledge creation and the maximization of an organization's knowledge and data assets is a key strategic necessity. Furthermore, we are seeing more techniques such as business intelligence and business analytics which have their foundations in traditional data mining being

embraced by organizations in order to try to facilitate the discovery of novel and unique patterns in data that will lead to new knowledge and maximization of an organization's data assets. Full maximization of an organization's data assets however will not be realized until the people perspective is incorporated into these data mining techniques to enable the full potential of knowledge creation to occur and even this is not sufficient for the ensuing decision to be most suitable. However, in addition, to taking such a socio-technical perspective, it is imperative to follow a systematic process in the extraction and application of the data, information and knowledge obtained, if indeed the consequent decision making is to be superior. As data warehouses and databases increase in volume and data mining and business intelligence capabilities increase in sophistication organizations that combine superior people centric and technology centric and most important the confluence of these two; i.e., socio-technical centric perspectives for knowledge creation will indeed have a sustainable competitive advantage in the current knowledge economy.

CONCLUSIONS

Sustainable competitive advantage is dependent on building and exploiting core competencies (Newell et al., 2002). In order to sustain competitive advantage, resources which are idiosyncratic (and thus scarce), and difficult to transfer or replicate are required (Grant, 1991). A knowledge-based view of the firm identifies knowledge as the organizational asset that enables sustainable competitive advantage especially in hyper competitive environments (Wickramasinghe, 2003; Davenport & Prusak, 1999; Zack,1999). This is attributed to the fact that barriers exist regarding the transfer and replication of knowledge (Wickramasinghe, 2003); thus making knowledge and knowledge management of strategic significance (Kanter, 1999). The key to maximizing the knowledge asset is in finding novel and actionable patterns and continuously creating new knowledge thereby increasing the extant knowledge base of the organization. It only becomes truly possible to support both major types of knowledge creation scenarios and thereby realize the synergistic effect of the respective strengths of these approaches in enabling superior decision making by embracing a holistic perspective to data mining; namely incorporating the people and process perspectives..

REFERENCES

Alavi, M & Leidner, D. (2001). Review: Knowledge management and knowledge management systems: Conceptual foundations and research issues. *MIS Quarterly 25*(1), 107-136.

Alavi, M & Leidner, D. (2001). Review: Knowledge Management And Knowledge Management Systems: Conceptual Foundations And Research Issues. *MIS Quarterly, 25*(1), 107-136.

Award, E. & Ghaziri, H. (2004). *Knowledge management.* Upper Saddle River, NJ: Prentice Hall.

Becerra-Fernandez, I . et al. (2004). *Knowledge management.* Upper Saddle River, NJ: Prentice Hall.

Becerra-Fernandez, I., & Sabherwal, R. (2001, summer). Organizational knowledge management: A contingency Perspective. *Journal of Management Information Systems 18*(1), 23-55.

Bendoly, E. (2003). Theory and support for process frameworks of knowledge discovery and data mining from ERP systems. *Information & Management, 40,*639-647.

Boland, R. & Tenkasi, R. (1995). Perspective making perspective taking. *Organizational Science 6*(3), 50-372

Boyd JR COL USAF, (1976). Destruction and creation, in R Coram "*Boyd*". New York:Little, Brown & Co.

Chang Lee, K. et al. (2005). KMPI: Measuring knowledge management performance. *Information & Management, 42*(3), 469-482.

Choi, B., & Lee, H. (2003). An empirical investigation of KM styles and their effect on corporate performance. *Information & Management, 40* pp.403-417

Churchman, C. (1971). *The design of inquiring systems: Basic concepts of systems and organizations.* New York: Basic Books Inc.

Courtney, J. (2001). Decision making and knowledge management in inquiring organizations: Toward a new decision-making paradigm for DSS. D*ecision Support*

Systems Special Issue on Knowledge Management, 31, 17-38

Davenport,T. , & Grover, V. (2001). Knowledge management. *Journal of Management Information Systems, 18*(1), 3-4

Davenport, T., & Prusak,L. (1999). *Working knowledge.* Boston: Harvard Business School Press.

Drucker, P. (1993). *Post-capitalist society*. New York: Harper Collins.

Fayyad, Piatetsky-Shapiro, Smyth,(1996). From data mining to knowledge discovery: An overview. In Fayyad, Piatetsky-Shapiro, Smyth, Uthurusamy, *Advances in knowledge discovery and data mining.* Menlo Park, CA: AAAI Press / The MIT Press.

Grant, R. (1991). The resource-based theory of competitive advantage: Implications for strategy formulation. *California Management Review, 33*(3), Spring, 114-135

Holsapple,C., & Joshi, K. , (2002). Knowledge manipulation activities: Results of a Delphi study. *Information & Management, 39,* 477-419.

Kanter, J. (1999). Knowledge management practically speaking. *Information Systems Management, Fall.*

Malhotra, Y. (2001). Knowledge management and new organizational form. In Malhotra, Y., & Malhotra, Y. (Eds), *Knowledge Management and Virtual Organizations,* Hershey, PA: IGI Publishing.

Massey, A., Montoya-Weiss, M., & O'Driscoll, T. (2002). Knowledge management in pursuit of performance: Insights from Nortel Networks. *MIS Quarterly 26*(3), 269-289.

Newell, S., Robertson, M., Scarbrough, H., & Swan, J. (2002). *Managing knowledge work palgrave*. New York.

Nonaka, I & Nishiguchi, T. (2001). *Knowledge emergence.* Oxford: Oxford University Press.

Nonaka, I. (1994). A dynamic theory of organizational knowledge creation. *Organizational Science 5*,14-37.

Polyani, M. (1958). *Personal knowledge: Towards a post-critical philosophy.* Chicago: University Press.

Polyani, M. (1966). *The tacit dimension.* London: Routledge & Kegan Paul.

Schultze, U. D. Leidner (2002). Studying knowledge management in information systems research: Discourses and theoretical assumptions. *MIS Quarterly 26*(3), 212-242.

Spiegler, I. (2003). Technology and knowledge: Bridging a "generating" gap. *Information and Management, 40,* 533-539.

Swan, J., Scarbrough, H., & Preston, J. (1999). Knowledge management: The next fad to forget people? In *Proceedings of the 7th European Conference in Information Systems.*

von Lubitz, D. & Wickramasinghe, N. (2005a). Networkcentric healthcare: Outline of entry portal concept. (in press) *International Journal of Electronic Business Management.*

von Lubitz, D. & Wickramasinghe, N. (2005b) Creating germane knowledgein dynamic environments. (in press) *International Journal Innovation and Learning.*

von Lubitz, D. & Wickramasinghe, N. (2006). Technology and healthcare: The doctrine of networkcentric healthcare operations. *International Journal of Electronic Healthcare*

Wickramasinghe, N. (2006). Knowledge creation: A meta-framework. *International Journal of Innovation and Learning, 3*(5), 558-573

Wickramasinghe, N., et al. (2003). "Knowledge management and data mining: Strategic imperatives for healthcare. In *Proceedings of the 3rd Hospital of the Future Conference, Warwick.*

Wickramasinghe, N. (2003). Do we practice what we preach: Are knowledge management systems in practice truly reflective of knowledge management systems in theory? *Business Process Management Journal, 9*(3), 295-316.

Wickramasinghe, N. & Mills, G. (2001). MARS: The electronic medical record system -The core of the Kaiser galaxy. *International Journal Healthcare Technology Management, 3*(5/6), 406-423

Zack, M. (1999). *Knowledge and strategy.* Boston: Butterworth Heinemann.

KEY TERMS

Combination: Knowledge transfer mode that involves new explicit knowledge being derived from existing explicit knowledge.

Explicit Knowledge: Or factual knowledge i.e. "know what" (Cabena et al, 1998), represents knowledge that is well established and documented.

Externalization: Knowledge transfer mode that involves new explicit knowledge being derived from existing tacit knowledge.

Internalization: Knowledge transfer mode that involves new tacit knowledge being derived from existing explicit knowledge.

Knowledge Spiral: The process of transforming the form of knowledge, and thus increasing the extant knowledge base as well as the amount and utilization of the knowledge within the organization

OODA Loop: Developed by Boyd, this framework traverses the decision maker through the critical steps of observation, orientation decision and action. At all times the access and extraction of germane knowledge, pertinent information and relevant data are key.

Socialization: Knowledge transfer mode that involves new tacit knowledge being derived from existing tacit knowledge.

Tacit Knowledge: Or experiential knowledge i.e.," know how" (Cabena et al, 1998) represents knowledge that is gained through experience and through doing.

Physical Data Warehousing Design

Ladjel Bellatreche
Poitiers University, France

Mukesh Mohania
IBM India Research Lab, India

INTRODUCTION

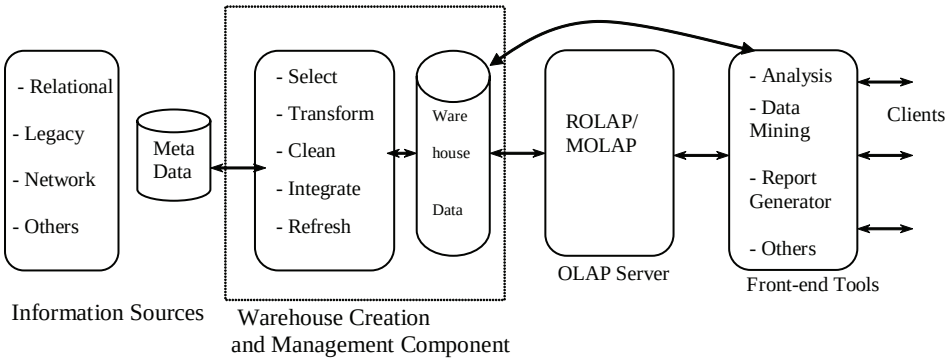
Recently, organizations have increasingly emphasized applications in which current and historical data are analyzed and explored comprehensively, identifying useful trends and creating summaries of the data in order to support high-level decision making. Every organization keeps accumulating data from different functional units, so that they can be analyzed (after integration), and important decisions can be made from the analytical results. Conceptually, a data warehouse is extremely simple. As popularized by Inmon (1992), it is a "subject-oriented, integrated, time-invariant, non-updatable collection of data used to support management decision-making processes and business intelligence". A data warehouse is a repository into which are placed all data relevant to the management of an organization and from which emerge the information and knowledge needed to effectively manage the organization. This management can be done using data-mining techniques, comparisons of historical data, and trend analysis. For such analysis, it is vital that (1) data should be accurate, complete, consistent, well defined, and time-stamped for informational purposes; and (2) data should follow business rules and satisfy integrity constraints. Designing a data warehouse is a lengthy, time-consuming, and iterative process. Due to the interactive nature of a data warehouse application, having fast query response time is a critical performance goal. Therefore, the physical design of a warehouse gets the lion's part of research done in the data warehousing area. Several techniques have been developed to meet the performance requirement of such an application, including materialized views, indexing techniques, partitioning and parallel processing, and so forth. Next, we briefly outline the architecture of a data warehousing system.

BACKGROUND

The conceptual architecture of a data warehousing system is shown in Figure 1. Data of a warehouse is extracted from operational databases (relational, object-oriented, or relational-object) and external sources (legacy data, other files formats) that may be distributed, autonomous, and heterogeneous. Before integrating this data into a warehouse, it should be cleaned to minimize errors and to fill in missing information, when possible, and transformed to reconcile semantic conflicts that can be found in the various sources. The cleaned and transformed data are integrated finally into a warehouse. Since the sources are updated periodically, it is necessary to refresh the warehouse. This component also is responsible for managing the warehouse data, creating indices on data tables, partitioning data, and updating meta-data. The warehouse data contain the detail data, summary data, consolidated data, and/or multi-dimensional data. The data typically are accessed and analyzed using tools, including OLAP query engines, data mining algorithms, information, visualization tools, statistical packages, and report generators.

The meta-data generally is held in a separate repository. The meta-data contain the informational data about the creation, management, and usage of tools (e.g., analytical tools, report writers, spreadsheets and data-mining tools) for analysis and informational purposes. Basically, the OLAP server interprets client queries (the client interacts with the front-end tools and passes these queries to the OLAP server) and converts them into complex SQL queries required to access the warehouse data. It also might access the data warehouse. It serves as a bridge between the users of the warehouse and the data contained in it. The warehouse data also are accessed by the OLAP server to present the data in a

Figure 1. A data warehousing architecture



multi-dimensional way to the front-end tools. Finally, the OLAP server passes the multi-dimensional views of data to the front-end tools, which format the data according to the client's requirements.

The warehouse data are typically modeled multi-dimensionally. The multi-dimensional data model has been proved to be the most suitable for OLAP applications. OLAP tools provide an environment for decision making and business modeling activities by supporting ad-hoc queries. There are two ways to implement a multi-dimensional data model: (1) by using the underlying relational architecture (star schemas, snowflake schemas) to project a pseudo-multi-dimensional model (example includes Informix Red Brick Warehouse); and (2) by using true multi-dimensional data structures such as, arrays (example includes Hyperion Essbase OLAP Server Hyperion). The advantage of MOLAP architecture is that it provides a direct multi-dimensional view of the data whereas the ROLAP architecture is just a multi-dimensional interface to relational data. On the other hand, the ROLAP architecture has two major advantages: (i) it can be used and easily integrated into other existing relational database systems; and (ii) relational data can be stored more efficiently than multi-dimensional data.

Data warehousing query operations include standard SQL operations, such as selection, projection, and join. In addition, it supports various extensions to aggregate functions, such as percentile functions (e.g., top 20th percentile of all products), rank functions (e.g., top 10 products), mean, mode, and median. One of the important extensions to the existing query language is to support multiple group-by, by defining roll-up, drill-down, and cube operators. Roll-up corresponds to doing further group-by on the same data object. Note that roll-up operator is order sensitive; that is, when it is defined in the extended SQL, the order of columns (attributes) matters. The function of a drill-down operation is the opposite of roll-up.

OLTP vs. OLAP

Relational database systems (RDBMS) are designed to record, retrieve, and manage large amounts of real-time transaction data and to keep the organization running by supporting daily business transactions (e.g., update transactions). These systems generally are tuned for a large community of users, and the user typically knows what is needed and generally accesses a small number of rows in a single transaction. Indeed, relational database systems are suited for robust and efficient Online Transaction Processing (OLTP) on operational data. Such OLTP applications have driven the growth of the DBMS industry in the past three decades and will doubtless continue to be important. One of the main objectives of relational systems is to maximize transaction throughput and minimize concurrency conflicts. However, these systems generally have limited decision support functions and do not extract all the necessary information required for faster, better, and intelligent decision making for the growth of an organization. For example, it is hard for an RDBMS to answer the following query: What are the supply patterns for product ABC in New Delhi in 2003, and how were they different from the year 2002? Therefore, it has become important to support analytical processing capabilities in organizations for (1) the efficient management of organizations, (2) effective marketing strategies, and (3) efficient and intelligent decision making. OLAP tools are well suited for complex data analysis, such as

Table 1. Comparison between OLTP and OLAP applications

Criteria	OLTP	OLAP
User	Clerk, IT Professional	Decision maker
Function	Day to day operations	Decision support
BD Design	Application-oriented	Subject-oriented
Data	Current	Historical, Consolidated
View	Detailed, flat relation	Summarized, multidimensional
Usage	Structured, Repetitive	Ad hoc
Unit of Work	Short, simple transaction	Complex query
Access	Read/Write	Append mostly
Records accessed	Tens	Millions
Users	Thousands	Tens
Size	MB-GB	GB-TB
Metric	Transaction throughput	Query throughput

multi-dimensional data analysis and to assist in decision support activities that access data from a separate repository called a data warehouse, which selects data from many operational legacies, and possibly heterogeneous data sources. The following table summarizes the differences between OLTP and OLAP.

MAIN THRUST

Decision-support systems demand speedy access to data, no matter how complex the query. To satisfy this objective, many optimization techniques exist in the literature. Most of these techniques are inherited from traditional relational database systems. Among them are materialized views (Bellatreche et al., 2000; Jixue et al., 2003; Mohania et al., 2000; Sanjay et al., 2000, 2001), indexing methods (Chaudhuri et al., 1999; Jügens et al., 2001; Stockinger et al., 2002), data partitioning (Bellatreche et al., 2000, 2002, 2004; Gopalkrishnan et al., 2000; Kalnis et al., 2002; Oracle, 2000; Sanjay et al., 2004), and parallel processing (Datta et al., 1998).

Materialized Views

Materialized views are used to precompute and store aggregated data, such as sum of sales. They also can be used to precompute joins with or without aggregations. So, materialized views are used to reduce the overhead associated with expensive joins or aggregations for a large or important class of queries. Two major problems related to materializing the views are (1) the view-maintenance problem, and (2) the view-selection problem. Data in the warehouse can be seen as materialized views generated from the underlying multiple data sources. Materialized views are used to speed up query processing on large amounts of data. These views need to be maintained in response to updates in the source data. This often is done using incremental techniques that access data from underlying sources. In a data-warehousing scenario, accessing base relations can be difficult; sometimes data sources may be unavailable, since these relations are distributed across different sources. For these reasons, the issue of self-maintainability of the view is an important issue in data warehousing. The warehouse views can be made self-maintainable by materializing some additional information, called auxiliary relations, derived from the intermediate results of the view computation. There are several algorithms, such as counting algorithm and exact-change algorithm, proposed in the literature for maintaining materialized views.

To answer the queries efficiently, a set of views that are closely related to the queries is materialized at the data warehouse. Note that not all possible views are materialized, as we are constrained by some resource like disk space, computation time, or maintenance cost. Hence, we need to select an appropriate set of views to materialize under some resource constraint. The view selection problem (VSP) consists in selecting a set of materialized views that satisfies the query response time under some resource constraints. All studies showed

that this problem is an NP-hard. Most of the proposed algorithms for the VSP are static. This is because each algorithm starts with a set of frequently asked queries (a priori known) and then selects a set of materialized views that minimize the query response time under some constraint. The selected materialized views will be a benefit only for a query belonging to the set of a priori known queries. The disadvantage of this kind of algorithm is that it contradicts the dynamic nature of decision support analysis. Especially for ad-hoc queries, where the expert user is looking for interesting trends in the data repository, the query pattern is difficult to predict.

Indexing Techniques

Indexing has been the foundation of performance tuning for databases for many years. It creates access structures that provide faster access to the base data relevant to the restriction criteria of queries. The size of the index structure should be manageable, so that benefits can be accrued by traversing such a structure. The traditional indexing strategies used in database systems do not work well in data warehousing environments. Most OLTP transactions typically access a small number of rows; most OLTP queries are point queries. B-trees, which are used in the most common relational database systems, are geared toward such point queries. They are well suited for accessing a small number of rows. An OLAP query typically accesses a large number of records for summarizing information. For example, an OLTP transaction would typically query for a customer who booked a flight on TWA 1234 on April 25, for instance; on the other hand, an OLAP query would be more like give me the number of customers who booked a flight on TWA 1234 in one month, for example. The second query would access more records that are of a type of range queries. B-tree indexing scheme is not suited to answer OLAP queries efficiently. An index can be a single-column or multi-column table (or view). An index either can be clustered or non-clustered. An index can be defined on one table (or view) or many tables using a join index. In the data warehouse context, when we talk about index, we refer to two different things: (1) indexing techniques and (2) the index selection problem. A number of indexing strategies have been suggested for data warehouses: value-list index, projection index, bitmap index, bit-sliced index, data index, join index, and star join index.

Data Partitioning and Parallel Processing

The data partitioning process decomposes large tables (fact tables, materialized views, indexes) into multiple (relatively) small tables by applying the selection operators. Consequently, the partitioning offers significant improvements in availability, administration, and table scan performance Oracle9i.

Two types of partitioning are possible to decompose a table: vertical and horizontal. In the vertical fragmentation, each partition consists of a set of columns of the original table. In the horizontal fragmentation, each partition consists of a set of rows of the original table. Two versions of horizontal fragmentation are available: primary horizontal fragmentation and derived horizontal fragmentation. The primary horizontal partitioning (HP) of a relation is performed using predicates that are defined on that table. On the other hand, the derived partitioning of a table results from predicates defined on another relation. In a context of ROLAP, the data partitioning is applied as follows (Bellatreche et al., 2002): it starts by fragmenting dimension tables, and then, by using the derived horizontal partitioning, it decomposes the fact table into several fact fragments. Moreover, by partitioning data of ROLAP schema (star schema or snowflake schema) among a set of processors, OLAP queries can be executed in a parallel, potentially achieving a linear speedup and thus significantly improving query response time (Datta et al., 1998). Therefore, the data partitioning and the parallel processing are two complementary techniques to achieve the reduction of query processing cost in data warehousing environments.

FUTURE TRENDS

It has been seen that many enterprises are moving toward building the Operational Data Store (ODS) solutions for real-time business analysis. The ODS gets data from one or more Enterprise Resource Planning (ERP) systems and keeps the most recent version of information for analysis rather than the history of data. Since the Client Relationship Management (CRM) offerings have evolved, there is a need for active integration of CRM with the ODS for real-time consulting and marketing (i.e., how to integrate ODS with CRM via messaging system for real-time business analysis).

Another trend that has been seen recently is that many enterprises are moving from data warehousing solutions to information integration (II). II refers to a category of middleware that lets applications access data as though they were in a single database, whether or not they are. It enables the integration of data and content sources to provide real-time read and write access in order to transform data for business analysis and data warehousing and to data placement for performance, currency, and availability. That is, we envisage that there will more focus on integrating the data and contents rather than only integrating structured data, as done in the data warehousing.

CONCLUSION

The data warehousing design is quite different from those of transactional database systems, commonly referred as Online Transaction Processing (OLTP) systems. A data warehouse tends to be extremely large, and the information in a warehouse usually is analyzed in a multi-dimensional way. The main objective of a data warehousing design is to facilitate the efficient query processing and maintenance of materialized views. For achieving this objective, it is important that the relevant data is materialized in the warehouse. Therefore, the problems of selecting materialized views and maintaining them are very important and have been addressed in this article. To further reduce the query processing cost, the data can be partitioned. That is, partitioning helps in reducing the irrelevant data access and eliminates costly joins. Further, partitioning at a finer granularity can increase the data access and processing cost. The third problem is index selection. We found that judicious index selection does reduce the cost of query processing, but we also showed that indices on materialized views improve the performance of queries even more. Since indices and materialized views compete for the same resource (storage), we found that it is possible to apply heuristics to distribute the storage space among materialized views and indices so as to efficiently execute queries and maintain materialized views and indexes.

It has been seen that enterprises are moving toward building the data warehousing and operational data store solutions.

REFERENCES

Bellatreche, L. et al. (2000). What can partitioning do for your data warehouses and data marts? *Proceedings of the International Database Engineering and Applications Symposium (IDEAS)*, Yokohoma, Japan.

Bellatreche, L. et al. (2002). PartJoin: An efficient storage and query execution for data warehouses. *Proceedings of the International Conference on Data Warehousing and Knowledge Discovery (DAWAK)*, Aix-en Provence, France.

Bellatreche, L. et al. (2004). Bringing together partitioning, materialized views and indexes to optimize performance of relational data warehouses. *Proceedings of the International Conference on Data Warehousing and Knowledge Discovery (DAWAK)*, Zaragoza, Spain.

Bellatreche, L., Karlapalem, K., & Schneider, M. (2000). On efficient storage space distribution among materialized views and indices in data warehousing environments. *Proceedings of the International Conference on Information and Knowledge Management (ACM CIKM)*, McLearn, VA, USA.

Chaudhuri, S., & Narasayya, V. (1999). Index merging. *Proceedings of the International Conference on Data Engineering (ICDE)*, Sydney, Australia.

Datta, A., Moon, B., & Thomas, H.M. (2000). A case for parallelism in data warehousing and olap. *Proceedings of the DEXA Workshop*, London, UK.

Gopalkrishnan, V., Li, Q., & Karlapalem, K. (2000). Efficient query processing with associated horizontal class partitioning in an object relational data warehousing environment. *Proceedings of the International Workshop on Design and Management of Data Warehouses*, Stockholm, Sweden.

Hyperion. (2000). Hyperion Essbase OLAP Server. *http://www.hyperion.com/*

Inmon, W.H. (1992). *Building the data warehouse*. John Wiley.

Jixue, L., Millist, W., Vincent, M., & Mohania, K. (2003). Maintaining views in object-relational databases. *Knowledge and Information Systems, 5*(1), 50-82.

Jürgens, M., Lenz, H. (2001). Tree based indexes versus bitmap indexes: A performance study. *International Journal of Cooperative Information Systems (IJCIS), 10*(3), 355-379.

Kalnis, P. et al. (2002). An adaptive peer-to-peer network for distributed caching of OLAP results. *Proceedings of the International Conference on Management of Data (ACM SIGMOD)*, Wisconsin, USA.

Mohania, M., & Kambayashi, Y. (2000). Making aggregate views self-maintainable. *Data & Knowledge Engineering, 32*(1), 87-109.

Oracle Corp. (2004). Oracle9i enterprise edition partitioning option. Retrieved from *http://otn.oracle.com/products/oracle9i/datasheets/partitioning.html*

Sanjay, A., Chaudhuri, S., & Narasayya, V.R. (2000). Automated selection of materialized views and indexes in Microsoft SQL server. *Proceedings of the 26th International Conference on Very Large Data Bases (VLDB'2000)*, Cairo, Egypt.

Sanjay, A., Chaudhuri, S., & Narasayya, V. (2001). Materialized view and index selection tool for Microsoft SQP server 2000. *Proceedings of the International Conference on Management of Data (ACM SIGMOD)*, CA.

Sanjay, A., Narasayya, V., & Yang, B. (2004). Integrating vertical and horizontal partitioning into automated physical database design. *Proceedings of the International Conference on Management of Data (ACM SIGMOD)*, Paris, France.

Stockinger, K. (2000). Bitmap indices for speeding up high-dimensional data analysis. *Proceedings of the International Conference Database and Expert Systems Applications (DEXA)*, London, UK.

KEY TERMS

Bitmap Index: Consists of a collection of bitmap vectors, each of which is created to represent each distinct value of the indexed column. A bit i in a bitmap vector, representing value x, is set to 1, if the record i in the indexed table contains x.

Cube: A multi-dimensional representation of data that can be viewed from different perspectives.

Data Warehouse: An integrated decision support database whose content is derived from the various operational databases. A subject-oriented, integrated, time-variant, non-updatable collection of data used in support of management decision-making processes.

Dimension: A business perspective useful for analyzing data. A dimension usually contains one or more hierarchies that can be used to drill up or down to different levels of detail.

Dimension Table: A table containing the data for one dimension within a star schema. The primary key is used to link to the fact table, and each level in the dimension has a corresponding field in the dimension table.

Fact Table: The central table in a star schema, containing the basic facts or measures of interest. Dimension fields are also included (as foreign keys) to link to each dimension table.

Horizontal Partitioning: Distributing the rows of a table into several separate tables.

Join Index: Built by translating restrictions on the column value of a dimension table to restrictions on a large fact table. The index is implemented using one of two representations: row id or bitmap, depending on the cardinality of the indexed column.

Legacy Data: Data that you already have and use. Most often, this takes the form of records in an existing database on a system in current use.

Measure: A numeric value stored in a fact table or cube. Typical examples include sales value, sales volume, price, stock, and headcount.

Star Schema: A simple database design in which dimensional data are separated from fact or event data. A dimensional model is another name for star schema.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 523-528, copyright 2005 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

Positive Unlabelled Learning for Document Classification

Xiao-Li Li
Institute for Infocomm Research, Singapore

See-Kiong Ng
Institute for Infocomm Research, Singapore

INTRODUCTION

In traditional supervised learning, a large number of labeled positive and negative examples are typically required to learn an accurate classifier. However, in practice, it is costly to obtain the class labels for large sets of training examples, and oftentimes the negative examples are lacking. Such practical considerations motivate the development of a new set of classification algorithms that can learn from a set of labeled positive examples *P* augmented with a set of unlabeled examples *U* (which contains both hidden positive and hidden negative examples). That is, we want to build a classifier using *P* and *U* in the absence of negative examples to classify the data in *U* as well as future test data. This problem is called the Positive Unlabelled learning problem or PU learning problem.

For instance, a computer scientist may want to build an up-to-date repository of machine learning (ML) papers. Ideally, one can start with an initial set of ML papers (e.g., a personal collection) and then use it to find other ML papers from related online journals or conference series, e.g., Artificial Intelligence journal, AAAI (National Conference on Artificial Intelligence), IJCAI (International Joint Conferences on Artificial Intelligence), SIGIR (ACM Conference on Research & Development on Information Retrieval), and KDD (ACM International Conference on Knowledge Discovery and Data Mining) etc. With the enormous volume of text documents on the Web, Internet news feeds, and digital libraries, finding those documents that are related to one's interest can be a real challenge.

In the application above, the class of documents that one is interested in is called the **positive documents** (i.e. the ML papers in the online sources). The set of known positive documents are represented as *P* (namely, the initial personal collection of ML papers). The unlabelled set *U* (papers from AAAI Proceedings etc) contains two groups of documents. One group contains documents of class *P*, which are the hidden positive documents in *U* (e.g., the ML papers in an AAAI Proceedings). The other group, which comprises the rest of the documents in *U*, are the **negative documents** (e.g., the non-ML papers in an AAAI Proceedings) since they do not belong to positive class. Given a positive set *P*, **PU learning** aims to identify a particular class *P* of documents from *U* or classify the future test set into positive and negative classes. Note that collecting unlabeled documents is normally easy and inexpensive, especially those involving online sources.

BACKGROUND

A theoretical study of PAC learning from positive and unlabeled examples under the statistical query model was first reported in (Denis, 1998). It assumes that the proportion of positive instances in the unlabeled set is known. Letouzey *et al.* (Letouzey, Denis, & Gilleron, 2000; Denis, Gilleron, & Letouzey, 2005) presented a learning algorithm based on a modified decision tree algorithm in this model. Muggleton (Muggleton, 1997) followed by studying the problem in a Bayesian framework where the distribution of functions and examples are assumed known. (Liu, Lee, Yu, & Li, 2002) reported sample complexity results and provided theoretical elaborations on how the problem may be solved.

A number of practical algorithms, S-EM, PEBL and Roc-SVM (Liu et al., 2002; Yu, Han, & Chang, 2002; Li & Liu, 2003) have also been proposed. They all conformed to the theoretical results in (Liu et al., 2002), following a two-step strategy: (1) identifying a set of reliable negative documents *RN* from the unlabeled set (called *strong negative documents* in PEBL), and (2) building a classifier using *P* (positive set), *RN* (negative set) and *U-RN*

(unlabelled set) through applying an existing learning algorithm (such as EM or SVM) once or iteratively. Their specific differences are described in the next subsection "Existing Techniques S-EM, PEBL and Roc-SVM".

Other related works include: Lee and Liu's weighted logistic regression technique (Lee & Liu, 2003) and Liu et al.'s biased SVM technique (Liu, Dai, Li, Lee, & Yu, 2003). Both required a performance criterion to determine the quality of the classifier. In (Fung, Yu, Lu, & Yu, 2006), a method called PN-SVM was proposed to deal with the case when the positive set is small where it assumes that the positive examples in *P* and the hidden positives in *U* were all generated from the same distribution. More recently, PU learning was used to identify unexpected instances in the test set (Li, Liu, & Ng, 2007b). PU learning was also useful for extracting relations, identifying user preferences and filtering junk email, etc (Agichtein, 2006; Deng, Chai, Tan, Ng, & Lee., 2004; Schneider, 2004; Zhang & Lee., 2005).

MAIN FOCUS

We will first introduce state-of-art techniques in PU learning for document classification, namely, S-EM (Liu et al., 2002), PEBL (Yu et al., 2002) and Roc-SVM (Li & Liu, 2003). Then, we will describe a document classification application which requires PU learning with only a small positive training set.

Existing Techniques S-EM, PEBL and Roc-SVM

As mentioned earlier, the existing techniques (S-EM, PEBL and Roc-SVM) all use a two-step strategy. We will focus on discussing the first step (negative example extraction) of the three methods (Spy, 1DNF, Rocchio respectively), since the second step of the three methods is essentially similar.

The main idea of S-EM is to use a spy technique to identify some *reliable negative documents* from the unlabeled set *U*. S-EM works by sending some "spy" documents from the positive set *P* to the unlabeled set *U*. The technique makes the following assumption: since the spy documents from *P* and the hidden positive documents in *U* are positive documents, the spy documents should behave identically to the hidden positive documents in *U* and can thus be used to reliably infer the behavior of the unknown positive documents.

S-EM randomly samples a set *S* of positive documents from *P* and puts them in *U*. Next, it runs the naïve Bayesian (NB) technique using the set $P - S$ as positive and the set $U \cup S$ as negative. The NB classifier is then applied to classify each document *d* in $U \cup S$, i.e., to assign it a probabilistic class label Pr(+|*d*), where "+" represents the positive class. Finally, it uses the probabilistic labels of the spies to decide which documents are most likely to be positive (the remaining documents are thus the reliable negative). S-EM sets a threshold using the probabilistic labels of spies which controls to extract most (85%) of spies from *U*, indicating that it can also extract out most of the other hidden positives from *U* since they behave identically.

In comparison, PEBL (Positive Example Based Learning) tries to extract reliable (strong) negative documents by using 1DNF method. Those documents in *U* that do not contain any positive features are regarded as reliable negative documents. 1DNF method first builds a positive feature set *PF* containing words that occur in the positive set *P* more frequently than in the unlabeled set *U*. Then it tries to filter out possible positive documents from *U*. A document in *U* that does not contain any positive feature in *PF* is regarded as a reliable negative document.

Unlike S-EM and PEBL, Roc-SVM performs negative data extraction using the Rocchio method. Roc-SVM uses the positive set *P* as positive training data and *U* as negative training data to build a Rocchio classifier where positive and negative prototype vectors $\vec{c}^{+}$ and $\vec{c}^{-}$ are constructed. In classification, for each document $\vec{d}'$ in unlabelled set U, it simply uses the cosine measure (Salton & McGill, 1986) to compute the similarity of $\vec{d}'$ with each prototype vector. The class whose prototype vector is more similar to $\vec{d}'$ is assigned to the document. Those documents classified as negative form the reliable negative set *RN*.

Let us compare the above three methods for extracting reliable negative documents. Although S-EM is not sensitive to noise, it can be problematic when the EM's assumptions (the data is generated by a mixture model, and there is a one-to-one correspondence between mixture components and classes) do not hold (Liu et al., 2002). PEBL is sensitive to the number of positive documents. When the positive data

is small, the results are often very poor (Li & Liu, 2003) (Fung et al., 2006). In contrast, Roc-SVM is robust and performs consistently well under a variety of conditions (Li & Liu, 2003; Fung et al., 2006). However, these current methods do not perform well when the size of the positive examples is small.

LPLP

In this section, we consider a problem of learning to classify documents with only a small positive training set. The work was motivated by a real-life business intelligence application of classifying web pages of product information. For example, a company that sells digital cameras may want to do a product comparison among the various cameras currently available in the market. One can first collect sample digital camera pages by crawling through all product pages from a consolidated e-commerce web site (e.g., amazon.com) and then hand-label those pages containing digital camera product information to construct the set *P* of positive examples. Next, to get more product information, one can then crawl through all the product pages from other web sites (e.g., cnet.com) as *U*. Ideally, PU learning techniques can then be applied to classify all pages in *U* into digital camera pages and non-digital camera pages. However, we found that while the digital camera product pages from two websites (say, amazon.com and cnet.com) do indeed share many similarities, they can also be quite distinct as the different web sites invariably present their products in different styles and have different focuses. As such, directly applying existing methods would give very poor results because 1) the small positive set *P* obtained from one site typically contains only tens of web pages (usually less than 30) and therefore do not adequately represent the whole positive class *P*, and 2) the features from the positive examples in *P* and the hidden positive examples in *U* are usually not generated from the same distribution because they were from different web sites.

The LPLP (Learning from Probabilistically Labeling Positive examples) method is designed to perform well for this task (Li, Liu, & Ng, 2007a). It consists of three steps. 1) Selecting a set of representative keywords from *P*; 2) Identifying likely positive set *LP* from *U* and probabilistically labeling the documents in *LP*; 3) Performing the classification algorithm. Note that LPLP is significantly different from current existing PU learning techniques since it focuses on extracting a set of likely positive documents from *U* instead of reliable negatives.

Selecting a Set of Representative Keywords From P

Notice that while the positive documents in *P* and the hidden positive documents in *U* were not drawn from the same distribution, they should still be similar in some underlying feature dimensions given that they belonged to the same class. For example, the digital camera pages from two different sites, say amazon.com and cnet.com, would share the representative keywords such as "resolution", "pixel", "lens", "shutter", "zoom", etc, even if their respective distributions may be quite different.

A set *RW* of representative keywords can be extracted from the positive set *P* which contains the top *k* words with the highest $s(w_i)$ scores. The scoring function $s()$ is based on TFIDF method (Salton & McGill, 1986) which gives high scores to those words that occur frequently in the positive set *P* and not in the whole corpus $P \cup U$ since *U* contains many other unrelated documents.

After removing the stop words and performing stemming (Salton & McGill, 1986), the LPLP algorithm computes the accumulated word frequency for each word feature w_i in the positive set *P*. The score of w_i is then computed by considering both w_i's probabilities of belonging to a positive class and its inverted document frequency in $P \cup U$. After ranking the scores for all the features into the rank list *L*, those word features with top *k* scores in *L* are stored into *RW*.

Identifying *LP* and Probabilistically Labeling the Documents

Once the set *RW* of representative keywords is determined, we can regard them together as a representative document (*rd*) of the positive class. The next objective is to identify the likely positive set *LP*, since the current positive set *P* is small.

The similarity of each document d_i in *U* with *rd* can be computed by using the cosine similarity metric (Salton & McGill, 1986). For any document d_i in *U*, if $sim(rd, d_i)>0$, it is stored into a set *LP* assigned with a probability $Pr(+|d_i)$ based on the ratio of the similarity $sim(rd, d_i)$ and *m* (the maximal similarity). Otherwise,

d_i is included in *RU* instead. The documents in *RU* have zero similarity with *rd* and can be considered as a "purer" negative set than *U*.

The hidden positive examples in *LP* will be assigned high probabilities $Pr(+|d_i)$ while the negative examples in *LP* will be assigned very low probabilities $Pr(+|d_i)$. This is because the representative keywords in *RW* were chosen based on those words that occurred frequently in *P* but not in the whole corpus $P \cup U$. As such, the hidden positive examples in *LP* should also contain many of the features in *RW* while the negative examples in *LP* would contain few of the features in *RW*.

Performing the Classification Algorithm

Finally, the proposed LPLP algorithm will build a NB classifier using *RU* as negative training set. As the positive set *PS*, there are two possible choices: either (1) combine *LP* and *P* as *PS*, or (2) use only *LP* as *PS*, depending on the situation. The initial NB classifier is then applied to the documents in ($LP \cup RU$) to obtain the posterior probabilities ($Pr(+|d_i)$ and $Pr(-|d_i)$) for each document. An EM algorithm can then iteratively employ the revised posterior probabilities to build a new NB classifier until the parameters of the NB classifier converge.

The LPLP algorithm was evaluated with experiments performed using actual product Web pages (Notebook, Digital Camera, Mobile Phone, Printer and TV) collected from 5 commercial Web sites: Amazon, CNet, PCMag, J&R and ZDnet. As expected, LPLP was found to perform significantly better than the three existing methods S-EM, PEBL and Roc-SVM. Its ability to handle probabilistic labels makes it better equipped to take advantage of the probabilistic *LP* set than the SVM-based approaches. As for the choices of two different *PS* sets, if there were only a small number of positive documents available, it was found that using combined *PS* for constructing the classifier is better since likely positive set *LP* can help represent the whole positive class better. Interestingly, if there is a large number of positive documents, using only *LP* is superior (Li et al., 2007a).

FUTURE TRENDS

We have described four techniques S-EM, PEBL, Roc-SVM, and LPLP that can be used in PU learning for document classification. One important empirical observation is that when the positive set is sufficiently large, all the PU learning techniques can achieve good classification results. However, when a small positive set is available, only the LPLP method can perform well. PU learning with small positive set *P* is a challenging problem since the small positive examples in *P* does not adequately represent the documents of whole positive class.

Current search engines are keyword-based instead of example-based, assuming that a user can always articulate what he/she is looking for with a few keywords. For complex information retrieval tasks, it may be useful for a user to provide a few positive examples that he/she is interested in. Given that the positive examples would provide more descriptive power than the keywords, PU learning with small positive set could potentially improve the accuracy of search engines further. This is one of the future directions of PU learning.

CONCLUSION

We have presented the PU learning problem and some existing techniques for it. Particularly, S-EM, PEBL and Roc-SVM, have been proposed to address the lack of negative examples by attempting to identify potential negative examples from the unlabelled set *U*. However, such PU learning methods would not work well in some real-world classification applications where the size of positive examples is small.

LPLP method was designed to address this oft-overlooked issue. Instead of identifying a set of reliable negative documents from the unlabeled set *U*, LPLP focuses on extracting a set of likely positive documents from *U*. This addresses the limited size of *P* and its potential distribution differences with the overall positive examples. With the augmentation of the extracted probabilistic *LP* set, the LPLP algorithm can build a much more robust classifier than those algorithms that rely only on the small positive set *P*.

REFERENCES

Agichtein, E. (2006). Confidence Estimation Methods for Partially Supervised Relation Extraction. Paper presented at the Proceedings of the SIAM International Confence on Data Mining (SDM06).

Deng, L., Chai, X., Tan, Q., Ng, W., & Lee., D. L. (2004). Spying Out Real User Preferences for Metasearch Engine Personalization. Paper presented at the Proceedings of the Workshop on WebKDD.

Denis, F. (1998). PAC Learning from Positive Statistical Queries, Proceedings of the 9th International Conference on Algorithmic Learning Theory (pp. 112-126).

Denis, F., Gilleron, R., & Letouzey, F. (2005). Learning from positive and unlabeled examples. Theoretical Computer Science, 348(1), 70-83.

Fung, G. P. C., Yu, J. X., Lu, H., & Yu, P. S. (2006). Text Classification without Negative Examples Revisit. IEEE Transactions on Knowledge and Data Engineering, 18(1), 6-20.

Lee, W. S., & Liu, B. (2003). Learning with Positive and Unlabeled Examples Using Weighted Logistic Regression, Proceedings of the Twentieth International Conference on Machine Learning (pp. 448-455).

Letouzey, F., Denis, F., & Gilleron, R. (2000). Learning from positive and unlabeled examples. Paper presented at the Proceedings of the Eleventh International Conference, ALT 2000, Sydney, Australia.

Li, X.-L., & Liu, B. (2003). Learning to Classify Texts Using Positive and Unlabeled Data. Paper presented at the Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico.

Li, X.-L., Liu, B., & Ng, S.-K. (2007a). Learning to Classify Documents with Only a Small Positive Training Set. Paper presented at the Proceedings of the 18th European Conference on Machine Learning (ECML) Poland.

Li, X.-L., Liu, B., & Ng, S.-K. (2007b). Learning to Identify Unexpected Instances in the Test Set. Paper presented at the Proceedings of Twentieth International Joint Conference on Artificial Intelligence, India.

Liu, B., Dai, Y., Li, X., Lee, W. S., & Yu, P. S. (2003). Building Text Classifiers Using Positive and Unlabeled Examples. Paper presented at the Proceedings of the Third IEEE International Conference on Data Mining (ICDM'03), Florida, USA.

Liu, B., Lee, W. S., Yu, P. S., & Li, X.-L. (2002). Partially Supervised Classification of Text Documents, Proceedings of the Nineteenth International Conference on Machine Learning (pp. 387-394).

Muggleton, S. (1997). Learning from Positive Data, Proceedings of the sixth International Workshop on Inductive Logic Programming (pp. 358-376): Springer-Verlag.

Salton, G., & McGill, M. J. (1986). Introduction to Modern Information Retrieval: McGraw-Hill, Inc.

Schneider, K.-M. (2004). Learning to Filter Junk E-Mail from Positive and Unlabeled Examples. Paper presented at the Proceedings of the IJCNLP.

Yu, H., Han, J., & Chang, K. C.-C. (2002). PEBL: Positive Example Based Learning for Web Page Classification Using SVM. Paper presented at the Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Alberta, Canada.

Zhang, D., & Lee., W. S. (2005). A Simple Probabilistic Approach to Learning from Positive and Unlabeled Examples. Paper presented at the Proceedings of the 5th Annual UK Workshop on Computational Intelligence.

Key Terms

1DNF: A technique is used to extract the strong negative documents that do not contain any positive feature.

Likely Positive Set: Based on the representative document (*rd*) from the positive set *P*, a set *LP* of likely positive documents can be extracted out from unlabelled set *U*, by computing the similarities between the documents in *U* and *rd*.

Positive Set: A set *P* of positive documents which we are interested in. It is possible that *P* contains the documents from multiple classes.

PU Learning: Given *P* and *U*, a classifier is built to identify positive documents in *U* or in a separate test set *T*. In other words, the objective of PU learning is to accurately classify the documents in *U* or *T* into documents from *P* and documents not from *P*.

Spy: S-EM works by sending some "spy" documents from the positive set *P* to the unlabeled set *U*. Since spy documents should behave identically to the hidden positive documents in *U* and hence allows to reliably infer the behavior of the unknown positive documents.

Two-Step Strategy of PU Learning: (1) given positive set *P* and unlabelled set *U*, the first step is to identify a set of reliable negative documents from the unlabeled set *U*; and second step is to build a classifier using EM or SVM once or iteratively.

Unlabeled Set: A set *U* of the mixed set which contains both positive documents from positive class *P* and also other types of negative documents.

Predicting Resource Usage for Capital Efficient Marketing

D. R. Mani
Massachusetts Institute of Technology and Harvard University, USA

Andrew L. Betz
Progressive Insurance, USA

James H. Drew
Verizon Laboratories, USA

INTRODUCTION

A structural conflict exists in businesses which sell services whose production costs are discontinuous and whose consumption is continuous but variable. A classic example is in businesses where capital-intensive infrastructure is necessary for provisioning service, but the capacity resulting from capital outlay is not always fully and efficiently utilized. Marketing departments focus on initiatives which increase infrastructure usage to improve both customer retention and on-going revenue. Engineering and operations departments focus on the cost of service provision to improve the capital efficiency of revenue dollar received. Consequently, a marketing initiative to increase infrastructure usage may be resisted by engineering if its introduction would require great capital expense to accommodate that increased usage. This conflict is exacerbated when a usage-enhancing initiative tends to increase usage variability so that capital expenditures are triggered with only small increases in total usage.

A data warehouse whose contents encompass both these organizational functions has the potential to mediate this conflict, and data mining can be the tool for this mediation. Marketing databases typically have customer data on rate plans, usage and past response to marketing promotions. Engineering databases generally record infrastructure locations, usages and capacities. Other information is often available from both general domains to allow for the aggregation, or clustering of customer types, rate plans and marketing promotions so that marketing proposals and their consequences can be systematically evaluated to aid in decision making. These databases generally contain such voluminous or complicated data that classical data analysis tools are inadequate. In this chapter, we look at a case study where data mining is applied to predicting capital-intensive resource or infrastructure usage, with the goal of guiding marketing decisions to enable capital efficient marketing. Although the data mining models developed in this chapter do not provide conclusive positions on specific marketing initiatives and their engineering consequences, the usage revenues and infrastructure performance predicted by these models provide systematic, sound, and quantitative input for making balanced and cost-effective business decisions.

BACKGROUND

In this business context, applying data mining (Berry and Linoff, 2004; Abramowicz and Zurada, 2000; Han and Kamber, 2006; Kantardzic and Zurada, 2005) to capital efficient marketing is illustrated here by a study from wireless telephony (Green, 2000) where marketing plans introduced to utilize excess off-peak network capacity (see Figure 1) could potentially result in requiring fresh capital outlays by indirectly driving peak demand to levels beyond current capacity.

We specifically consider marketing initiatives—e.g., specific rate plans with free nights and weekends—that are aimed at stimulating off-peak usage. Given a rate plan with a fixed peak minute allowance, availability of extra off-peak minutes could potentially increase peak usage. The quantification of this effect is complicated by the corporate reality of myriad rate plans and geographically extensive and complicated peak usage patterns. In this study, we use data mining methods to analyze customer, call detail, rate plan and cell-site location data to predict the effect of marketing initiatives on busy hour network utilization. This will enable forecasting

Figure 2. Flowchart describing data sources and data mining operations used in predicting busy hour impact of marketing initiatives

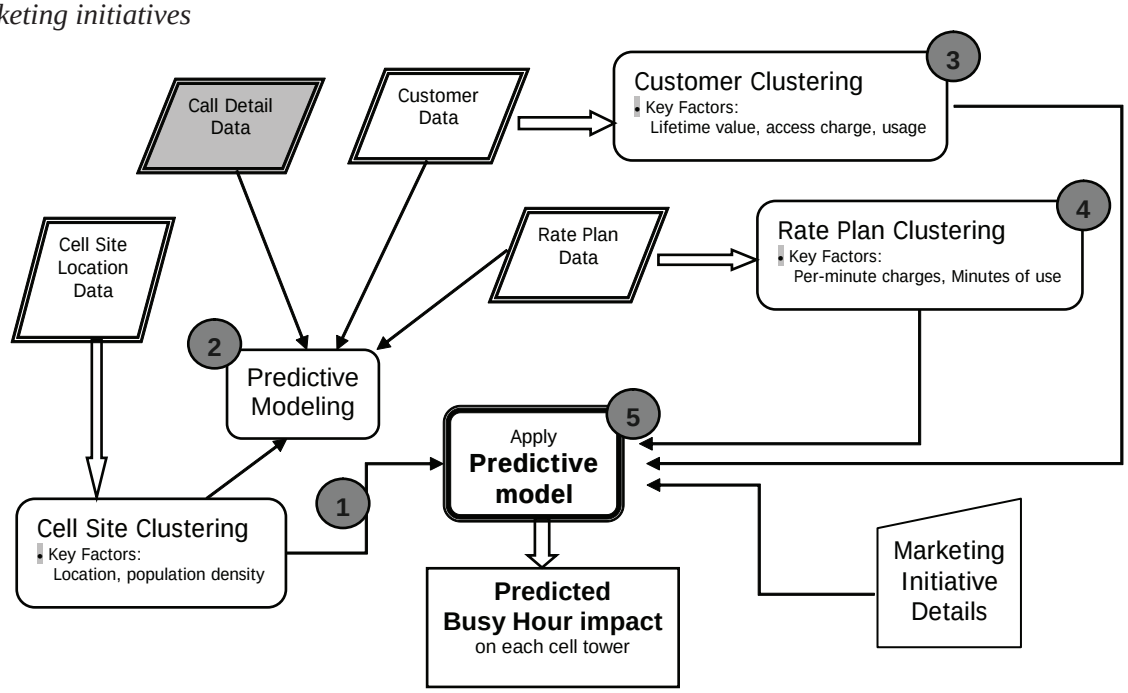
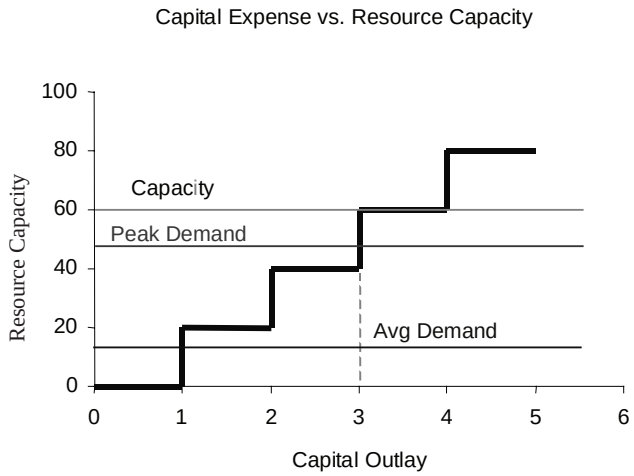


Figure 1. Marketing initiatives to boost average demand can indirectly increase peak demand to beyond capacity.



network cost of service for marketing initiatives thereby leading to optimization of capital outlay.

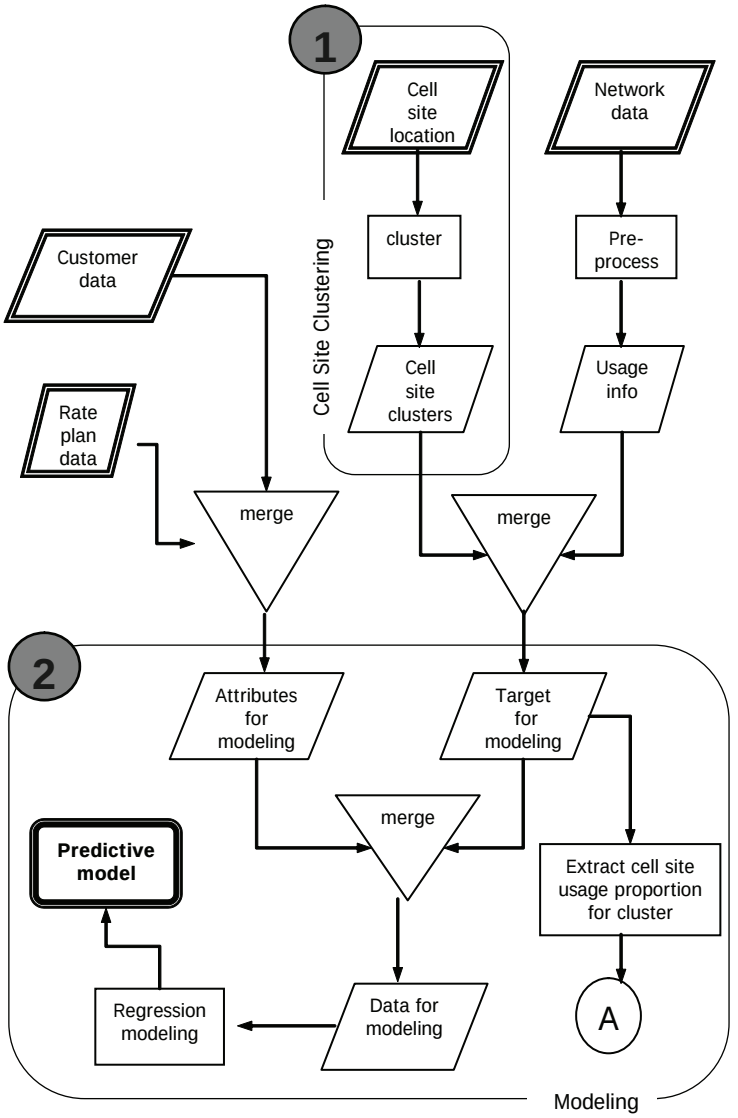
MAIN THRUST OF THE CHAPTER

Ideally, the capital cost of a marketing initiative is obtained by determining the existing capacity, the increased capacity required under the new initiative, and then factoring the cost of the additional capital; and data for a study like this would come from a corporate data warehouse (Berson and Smith, 1997) that integrates data from relevant sources. Unfortunately, such detailed cost data are not available in most corporations and businesses—in fact in many situations, the connection between promotional marketing initiatives and capital cost is not even recognized. In this case study, we therefore need to assemble relevant data from different and disparate sources in order to predict the busy hour impact of marketing initiatives.

Data

The parallelograms in the flowchart in Figure 2 indicate essential data sources for linking marketing initiatives to busy hour usage. *Customer Data* characterizes the customer by indicating a customer's mobile phone number(s), lifetime value, access charge, subscribed rate plan, peak and off-peak minutes used. *Rate Plan Data* provides details for a given rate plan including monthly charges, allowed peak, off-peak and weekend minutes of use, per-minute charges for excess use, long distance and roaming charges, etc. *Call Detail Data* provides, for every call placed in a given time period, the originating and terminating phone numbers

Figure 3. Steps 1 and 2 in the data mining process outlined in Figure 2



(and hence originating and terminating customers), cell sites used in handling the call, call duration and other call details. *Cell Site Location Data* indicates the geographic location of cell sites, capacity of each cell site, and details about the physical and electromagnetic configuration of the radio towers.

Data Mining Process

Figure 2 provides an overview of the analysis and data mining process. The numbered processes are described in more detail to illustrate how the various components are integrated into an exploratory tool that allows marketers and network engineers to evaluate the effect of proposed initiatives.

Cell-Site Clustering

Clustering cell sites using the geographic location (latitude, longitude) results in cell site clusters that capture the underlying population density, with cluster area generally inversely proportional to population. This is a natural consequence of the fact that heavily populated urban areas tend to have more cell towers to cover the large call volumes and provide good signal coverage. The flowchart for cell site clustering is included in

Figure 3 with results of *k*-means clustering (Hastie, Tibshirani and Friedman, 2001; Cios et. al., 2007) for the San Francisco area shown in Figure 4. The cell sites in the area are grouped into four clusters, each cluster approximately circumscribed by an oval.

Predictive Modeling

The predictive modeling stage, shown in Figure 3, merges customer and rate plan data to provide the data attributes (features) used for modeling. The target for modeling is obtained by combining cell site clusters with network usage information. Note that the sheer volume of call detail data makes its summarization and merging with other customer and operational data a daunting one. See, for example, Berry and Linoff (2000) for a discussion. The resulting dataset has, for every customer, related customer characteristics and rate plan details, matched up with that customer's actual network usage and the cell sites providing service for that customer. This data can then be used to build a predictive model. Feature selection (Liu and Motoda, 1998) is performed based on correlation to target, with grouped interactions taken into account. The actual predictive model is based on linear regression (Hastie, Tibshirani and Friedman, 2001; Hand, Mannila and Smyth, 2001)—which for this application performs similar to neural network models (Haykin, 1998; Duda, Hart and Stork, 2000). Note that the sheer number of customer and rate plan characteristics requires the variable reduction capabilities of a data mining solution.

For each of the 4 cell site clusters shown in Figure 4, Figure 5 shows the actual versus predicted busy hour usage, for calls initiated during a specific month in the San Francisco region. With statistically significant R^2 correlation values ranging from about 0.31 to 0.67, the models, while not perfect, are quite good at predicting cluster-level usage.

Figure 4. Clustering cell sites in the San Francisco area, based on geographic position. Each spot represents a cell site, with the ovals showing approximate location of cell site clusters

Customer and Rate Plan Clustering

Using *k*-means clustering to cluster customers based on their attributes, and rate plans based on their salient features, results in categorizing customers and rate plans into a small number of groups. Without such clustering, the complex data combinations and analyses needed to predict busy hour usage will result in very little predictive power. Clusters ameliorate the situation by providing a small set of representative cases to use in the prediction.

As in cell site clustering, based on empirical exploration, we decide on using four clusters for both customers and rate plans. Results of the clustering for customers and applicable rate plans in the San Francisco area are shown in Figure 6 and 7.

Integration and Application of the Predictive Model

Stringing all the pieces together, we can start with a proposed rate plan and an inwards projection of the expected number of customers who would subscribe to the plan, and predict the busy hour usage on targeted cell sites. The process is outlined in Figure 8. Inputs labeled A, B and C come from the respective flowcharts in Figures 2, 6 and 7.

Validation

Directly evaluating and validating the analytic model developed would require data summarizing capital cost of a marketing initiative. This requires determining the existing capacity, the increased capacity required under the new initiative, and then factoring the cost of the additional capital. Unfortunately, as mentioned earlier,

Figure 5. Regression modeling results showing predicted versus actual busy hour usage for the cell site clusters shown in Figure 4

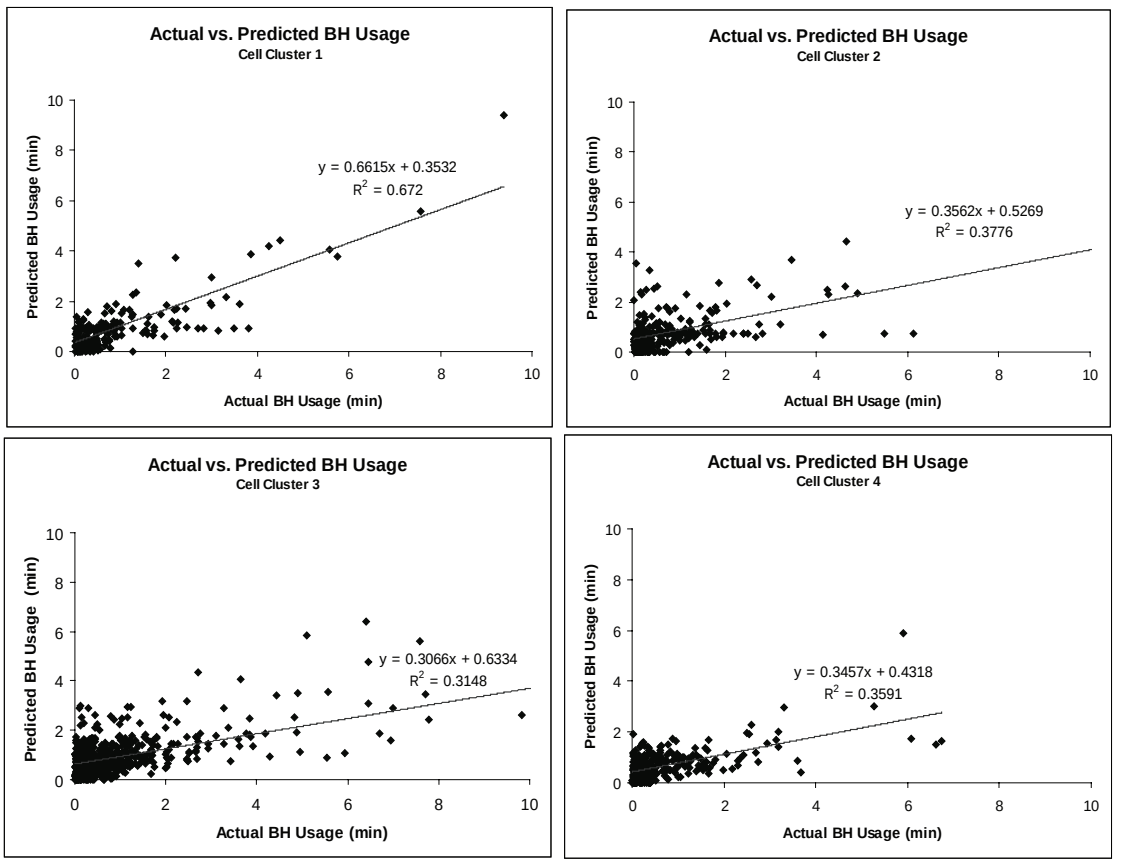


Figure 6. Customer clustering flowchart and results of customer clustering. The key attributes identifying the four clusters are access charge (ACC_CHRG), lifetime value (LTV_PCT), peak minutes of use (PEAK_MOU) and total calls (TOT_CALL)

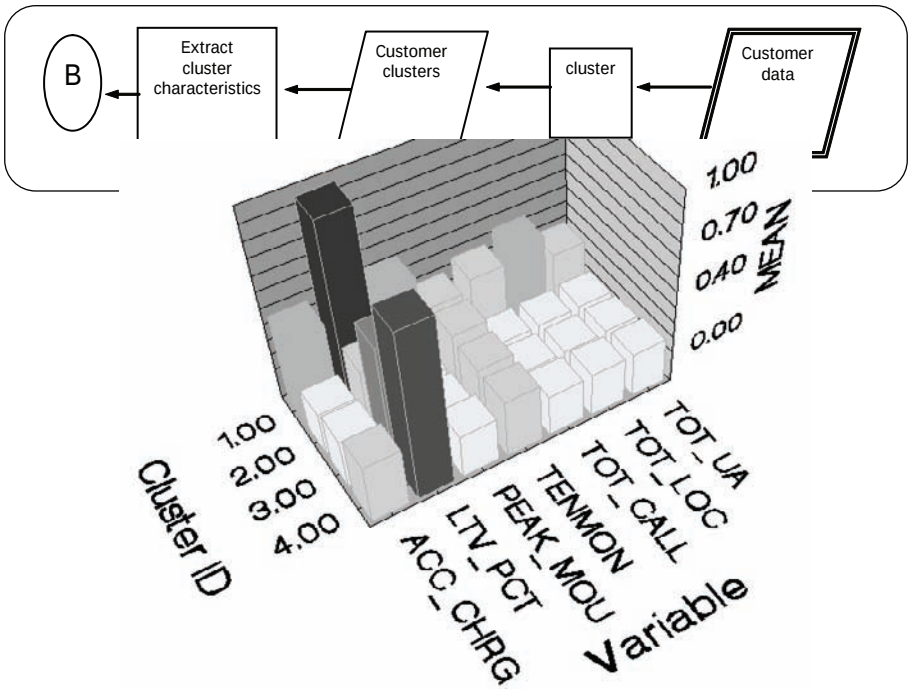


Figure 7. Rate plan clustering flowchart and clustering results. The driving factor that distinguish rate plan clusters include per-minute charges (intersystem, peak and off-peak), and minutes of use

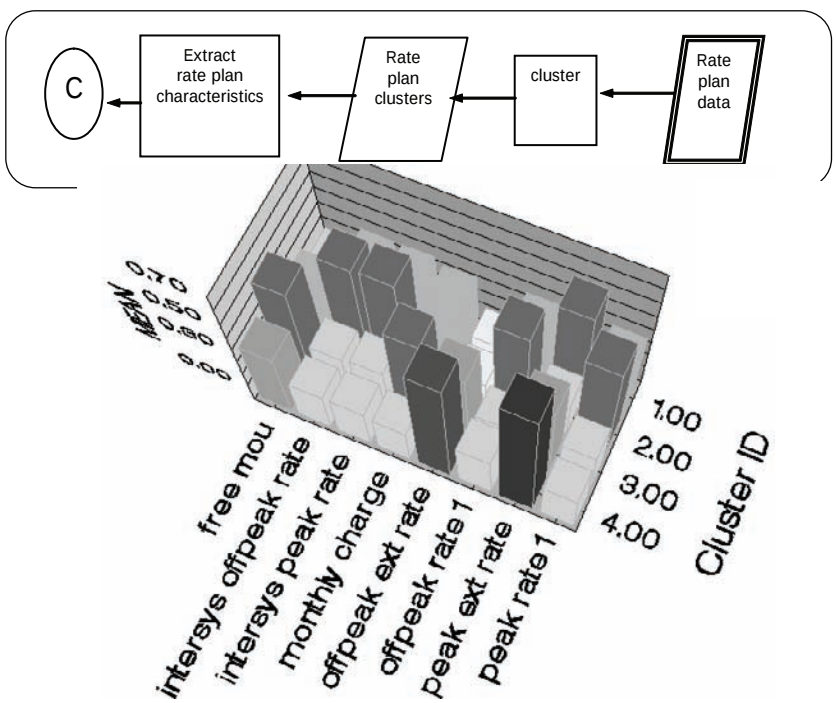
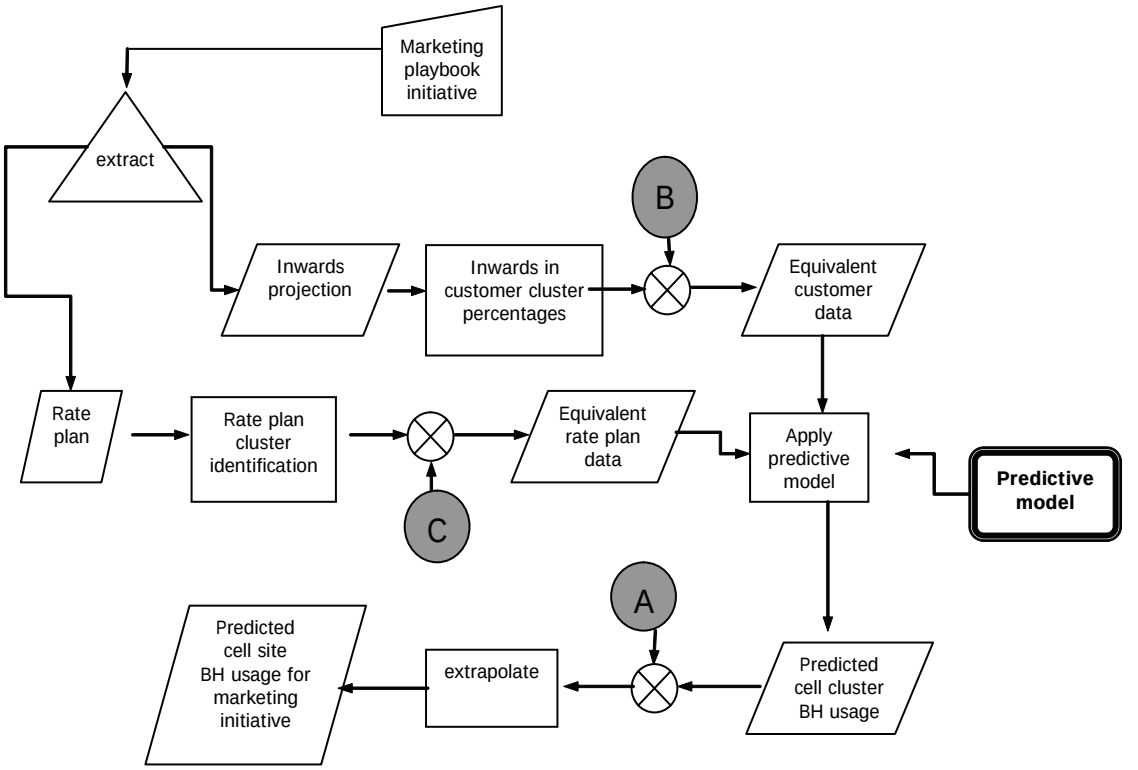


Figure 8. Predicting busy hour usage at a cell site for a given marketing initiative



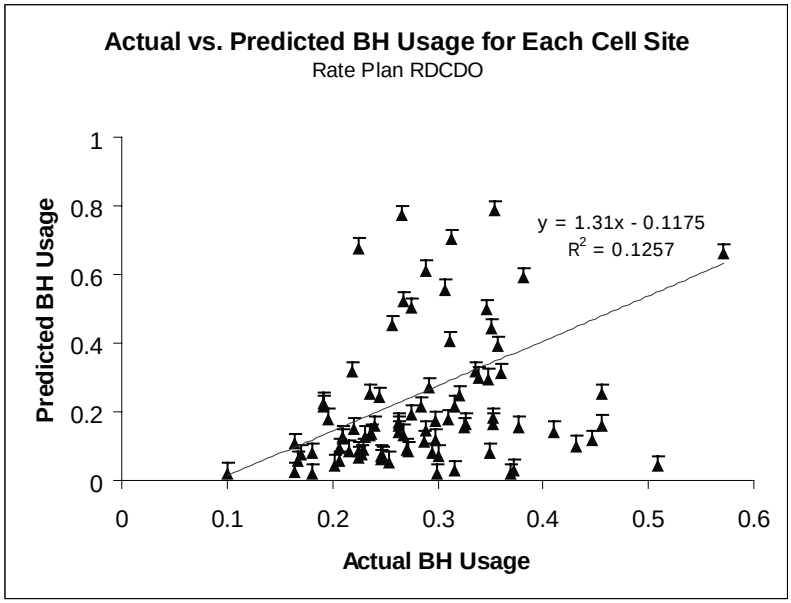
such detailed cost data are generally unavailable. We therefore validate the analytic model by estimating its impact and comparing the estimate with known parameters.

We began this estimation by identifying the ten most frequent rate plans in the data warehouse. From this list we selected one rate plan for validation. Based on marketing area and geographic factors, we assume an equal inwards projection for each customer cluster. Of course, our approach allows for, but does not require, different inward projections for the customer clusters. This flexibility could be useful, for example, when price elasticity data suggest inward projection differences by cluster.

Starting at the cluster level, we applied the predictive model to estimate the average busy hour usage for each customer on each cell tower. These cluster-level predictions were disaggregated to the cellular-tower level by assigning busy-hour minutes proportionate to total minutes for each cell tower within the respective cellular-tower cluster. Following this, we then calculated the actual busy hour usage per customer of that rate plan across the millions of call records. In Figure 9, a scatter plot of actual busy hour usage against predicted busy hour usage, with individual cell tower now the unit of analysis, reveals an R^2 correlation measure of 0.13.

The estimated model accuracy dropped from R^2s in the mid 0.30s for cluster-level data (Figure 5) to about 0.13 when the data were disaggregated to the cellular tower level (Figure 9). In spite of the relatively low R^2 value, the correlation is statistically significant, indicating that this approach can make contributions to the capital estimates of marketing initiatives. However, the model accuracy on disaggregated data was certainly lower than the effects observed at the cluster level. The reasons for this loss in accuracy could probably be attributed to the fine-grained disaggregation, the large variability among the cell sites in terms of busy hour usage, the proportionality assumption made in disaggregating, and model sensitivity to inward projections. These data point up both an opportunity (that rate plans can be intelligently targeted to specific locations with excess capacity) and a threat (that high busy hour volatility would lead engineering to be cautious in allowing usage-increasing plans).

Figure 9. Scatter plot showing actual versus predicted busy hour usage for each cell site, for a specific rate plan



Business Insights

In the context of the analytic model, the data can also reveal some interesting business insights. First, observe the customer lifetime value density plots as a function of strain they place on the network. The left panel in Figure 10 shows LTV density for customers with below average total usage and below average busy-hour usage. The panel on the right shows LTV density for customers with above average total usage and above average busy hour usage. Although the predominant thinking in marketing circles is that "higher LTV is always better," the data suggest this reasoning should be tempered by whether the added value in revenue offsets the disproportionate strain on network resources. This is the basis for a fundamental tension between marketing and engineering functions in large businesses.

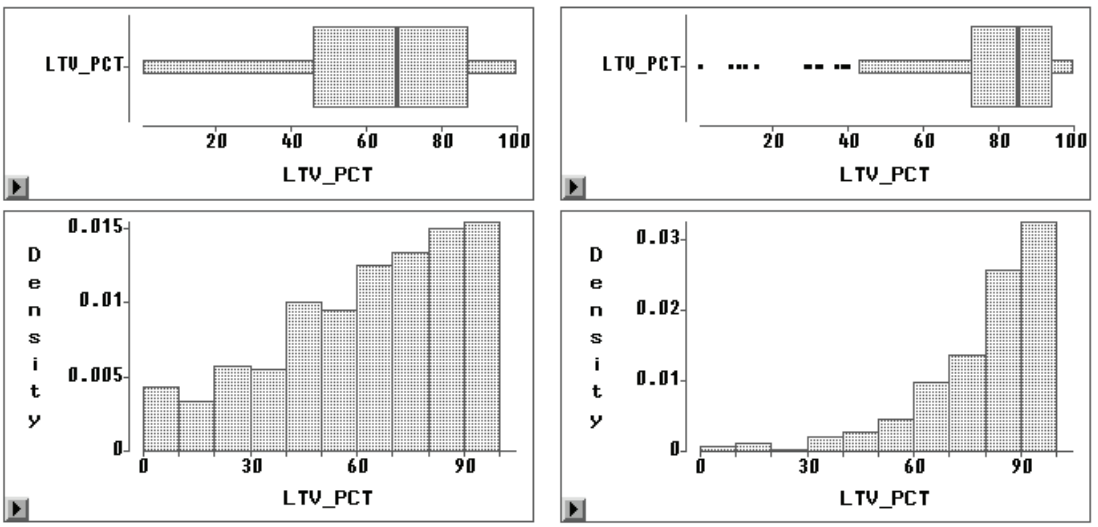
Looking at busy hour impact by customer cluster and rate plan cluster is also informative, as shown in Figure 11. For example, if we define "heavy BH users" as customers who are above average in total minutes as well as busy hour minutes, we can see main effect differences across the customer clusters (Figure 11a).

This is not entirely surprising, as we have already seen that LTV was a major determinant of customer cluster and heavy BH customers also skewed towards having higher LTV. There was, however, an unexpected crossover interaction of rate plan cluster by customer cluster when "heavy BH users" was the target (Figure 11b). The implication is that controlling for revenue, certain rate plan types are more network-friendly depending on the customer cluster under study. Capital-conscious marketers could in theory tailor their rate plans to minimize network impact by more precisely tuning rate plans to customer segments.

FUTURE TRENDS

The proof of concept case study sketched above describes an analytical effort to optimize capital costs of marketing programs, while promoting customer satisfaction and retention by utilizing the business insights gained to tailor marketing programs to better match the needs and requirements of customers (see, for example, Lenskold, 2003). Even as this linkage of marketing innovation and engineering capacity will remain an important business goal well into the foreseeable future, there are practical aspects to the updating of the analysis and models as the industry evolves. Although rate plans, calling areas and customer

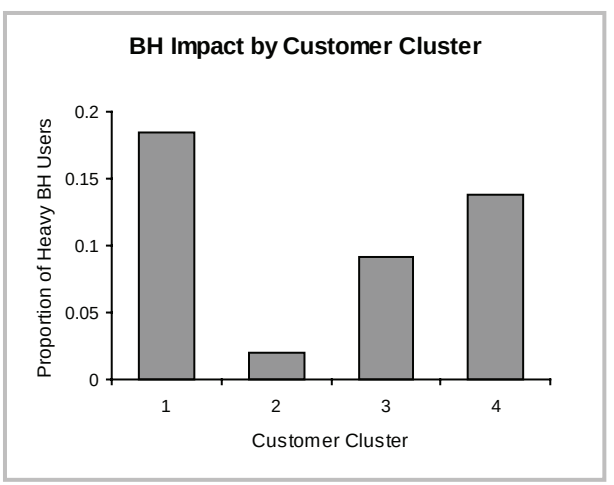
Figure 10. Life-time value (LTV) distributions for customers with below and above average usage and BH usage indicates that high-LTV customers also impact the network with high BH usage



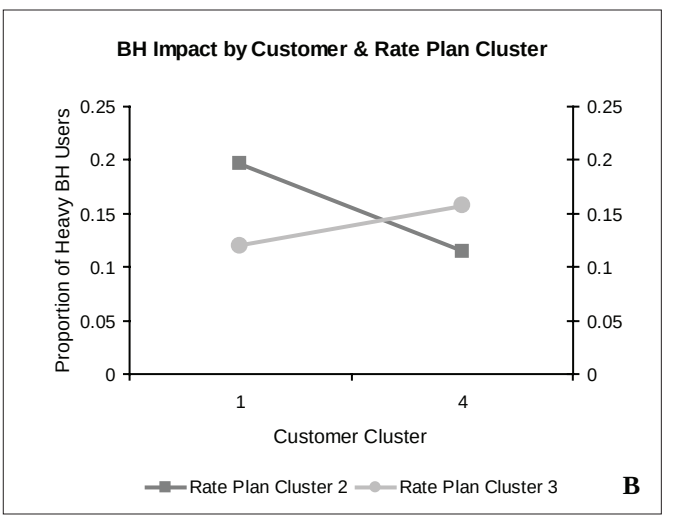
LTV distribution for customers with **less** than average usage and BH usage

LTV distribution for customers with **more** than average usage and BH usage

Figure 11. Exploring customer clusters and rate plan clusters in the context of BH usage shows that customers in a cluster who subscribe to a class of rate plans impact BH usage differently from customers in a different cluster



A



B

characteristics change continuously, the coefficients of the models above should change much less frequently. New plans with new characteristic levels appear, and customer characteristics change as cell phones become ubiquitous, but their effects can be predicted in the same way as the older characteristics. When the models' predictions become too inaccurate, as they eventually must, a series of changes, from simple to complex, can be built into the models. For instance, the intercept or offset of the usage model can easily be changed to accommodate more of the same mix of customers, and later, intercepts for individual clusters can be similarly tuned. This kind of simple modification may allow the essential models to be useful for long periods. Ultimately, though, we expect database systems to incorporate data mining engines (e.g., Oracle Data Mining, Oracle, 2007) and future software and customer relationship management applications to automatically incorporate such analysis to extract timely and optimal recommendations and business insights for maximizing business return on investment and customer satisfaction—leading to effective and profitable one-to-one customer relationship management (Greenberg, 2001; Brown, 2000; Gilmore and Pine, 2000).

CONCLUSION

We have made the general argument that a comprehensive company data warehouse and broad sets of models analyzed with modern data mining techniques can resolve tactical differences between different organizations within the company, and provide a systematic and sound basis for business decision making. The role of data mining in so doing has been illustrated here in mediating between the marketing and engineering functions in a wireless telephony company, where statistical models are used to target specific customer segments with rate plan promotions to increase overall usage (and hence retention) while more efficiently using, but not exceeding, network capacity. A further practical advantage of this data mining approach is that all of the customer groups, cell site locations and rate plan promotions are simplified through clustering to simplify their characteristic representation facilitating productive discussion among upper management strategists.

We have sketched several important practical business insights that come from this methodology. A basic concept to be demonstrated to upper management is that busy hour usage, the main driver of equipment capacity needs, varies greatly by customer segment, and by general cell site grouping (see Figure 9), and that usage can be differently volatile over site groupings (see Figure 5). These differences point out the potential need for targeting specific customer segments with specific rate plan promotions in specific locations. One interesting example is illustrated in Figure 11, where two customer segments respond differently to each of two candidate rate plans—one group decreasing its BH usage under one plan, while the other decreasing it under the other plan. This leads to the possibility that certain rate plans should be offered to specific customer groups in locations where there is little excess equipment capacity, but others can be offered with there is more slack in capacity.

There are, of course, many organizational, implementation and deployment issues associated with this integrated approach. All involved business functions must accept the validity of the modeling and its results, and this concurrence requires the active support of upper management overseeing each function. Secondly, these models should be repeatedly applied and occasionally rerun to assure their relevance in a dynamic business environment, as wireless telephony is in our illustration. Thirdly, provision should be made for capturing the sales and engineering consequences of the decisions made from this approach, to be built into future models. Despite these organizational challenges, business decision makers informed by these models may hopefully resolve a fundamental business rivalry.

REFERENCES

Abramowicz, W., & Zurada, J. (2000). *Knowledge discovery for business information systems*. Kluwer.

Berson, A., & Smith, S. (1997). *Data warehousing, data mining and OLAP*. McGraw-Hill.

Berry, M. A., & Linoff, G. S. (2000). *Mastering data mining: The art and science of customer relationship management*. Wiley.

Berry, M. A., & Linoff, G. S. (2004). *Data mining techniques: For marketing, sales, and customer relationship management, 2nd Ed.* Wiley.

Brown, S. A. (2000). *Customer relationship management: Linking people, process, and technology*. John Wiley.

Cios, K.J., Pedrycz, W., Swiniarski, R.W., & Kurgan, L.A. (2007) *Data mining: A knowledge discovery approach*. Springer.

Drew, J., Mani, D. R., Betz, A., & Datta, P. (2001). Targeting customer with statistical and data-mining techniques. *Journal of Services Research, 3*(3):205-219.

Duda, R. O., Hart, P. E., & Stork, D. E. (2000). *Pattern classification, 2nd Ed*. Wiley Interscience.

Gilmore, J., & Pine, J. (2000). *Markets of one*. Harvard Business School Press.

Green, J.H. (2000). *The Irwin handbook of telecommunications, 4th Ed*. McGraw-Hill.

Greenberg, P. (2001). *CRM at the speed of light: Capturing and keeping customers in Internet real time*. Mc-Graw Hill.

Han, J., & Kamber, M. (2006). *Data mining: Concepts and techniques, 2nd Edition*. Morgan Kaufmann.

Hand, D. J., Mannila, H., & Smyth, P. (2001). *Principles of data mining*. Bradford Books.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning: Data mining, inference, and prediction*. Springer.

Haykin, S. (1998). *Neural networks: A comprehensive foundation*. Prentice-Hall.

Kantardzic, M., & Zurada, J. (2005). *Next generation of data-mining applications*. Wiley-IEEE Press.

Lenskold, J. D. (2003). *Marketing ROI: The path to campaign, customer, and corporate profitability*. Mc-Graw Hill.

Liu, H, & Motoda, H. (1998). *Feature selection for knowledge discovery and data mining*. Kluwer.

Mani, D. R., Drew, J., Betz, A., & Datta, P. (1999). Statistics and data mining techniques for lifetime value modeling. *Proceedings of the Fifth ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. pp. 94-103. San Diego, CA.

Oracle. (2007). http://www.oracle.com/technology/products/bi/odm/index.html. Oracle Corporation.

KEY TERMS

Capital-Efficient Marketing: Marketing initiatives that explicitly take into account, and optimize if possible, the capital cost of provisioning service for the introduced initiative or promotion.

Marketing Plan: Also referred to as *marketing initiatives*, these are promotions and incentives, mediated through a *rate plan*—a mobile calling plan that specifies minute usage allowances at peak and off-peak times, monthly charge, cost of extra minutes, roaming and long-distance charges,etc.

Capacity: The maximum number of simultaneous calls that can be handled by the wireless network at any given time. Times during the day when call volumes are higher than average are referred to as *peak hours*; *off-peak hours* entail minimal call traffic—like nights and weekends.

Busy Hour: The hour at which a mobile telephone network handles the maximum call traffic in a 24 hour period. It is that hour during the day or night when the product of the average number of incoming calls and average call duration is at its maximum.

Inwards Projection: Estimating the expected number of customers that would sign on to a marketing promotion, based on customer characteristics and profiles.

Lifetime Value: A measure of the profit generating potential, or value, of a customer, and is a composite of expected tenure (how long the customer stays with the business provider) and expected revenue (how much does a customer spend with the business provider).

Predictive Modeling: Use of statistical or data mining methods to relate attributes (input features) to targets (outputs) using previously existing data for training, in such a manner that the target can be predicted for new data based on the input attributes alone.

***k*-Means Clustering:** A clustering method that groups items that are close together—based on a distance metric (like Euclidean distance)—to form clusters. The members in each of the resulting *k* clusters can be

described succinctly using the mean (or centroid) of the respective cluster.

Feature Selection: The process of identifying those input attributes that significantly contribute to building a predictive model for a specified output or target.

Preference Modeling and Mining for Personalization

Seung-won Hwang
Pohang University of Science and Technology (POSTECH), Korea

INTRODUCTION

As near-infinite amount of data are becoming accessible on the Web, it becomes more important to support intelligent personalized retrieval mechanisms, to help users identify the results of a manageable size satisfying user-specific needs. Example case studies include major search engines, such as Google and Yahoo, recently released personalized search, which adapts the ranking to the user-specific search context. Similarly, e-commerce sites, such as Amazon, are providing personalized product recommendation based on the purchase history and user browsing behaviors. To achieve this goal, it is important to model user preference and mine user preferences from user behaviors (e.g., click history) for personalization. In this article, we discuss recent efforts to extend mining research to preference and identify goals for the future works.

BACKGROUND

Traditional modeling for user preference can be categorized into (1) quantitative and (2) qualitative approaches. In the quantitative approach, given a set of data objects *D*, a utility function *F* assigns a numerical score *F(o)* for an object *o* in *D*. This utility score may aggregate scores on one (i.e., uni-attribute model) or more (i.e., multi-attribute model) *attributes* $F(a_1, \ldots, a_n)$, when $o = (a_1, \ldots, a_n)$. For instance, the utility of a house with *price = 100k* and *size = 100 square foot* can be quantified by a user-specific utility function, e.g., *F = size/price*, into a score, such that houses maximizing the scores, e.g., with largest size per unit price, can be retrieved.

Alternatively, in the qualitative approach, the preference on each object is stated in comparison to other objects. That is, given two objects *x* and *y*, instead of quantifying preferences into numerical scores, users simply state which one is more preferred, denoted as $x > y$ or $y > x$. Alternatively, users may state indifference $x \quad y$. Compared to quantitative modeling requiring users to quantify numerical scores of all objects, qualitative modeling is considered to be more intuitive to formulate (Payne, Bettman, & Johnson, 1993), while less efficient for evaluating the absolute utility of the specific object, as such evaluation requires relative comparisons to all other objects. Meanwhile, qualitative approach may also aggregate multiple orderings. Optimal results from such aggregation is formally defined as pareto-optimality as stated below.

Definition 1 (Pareto-optimality). A tuple *x* dominates another tuple *y* if and only if as $x > y$ or $x \quad y$ in all the given orderings.

MAIN FOCUS

Major components of enabling personalized retrieval can be identified as (1) preference modeling, (2) preference mining, and (3) personalization, each of which will be discussed in the following three subsections.

Preference Modeling

As discussed in the prior section, preferences are modeled typically as (1) quantitative utility function (Fishburn, 1970; Keeney & Raiffa, 1976) or (2) qualitative utility orderings (Payne et al., 1993). Personalization is guided by preferences represented in these two models, to retrieve ideal data results that maximize the utility. To maximize quantitative utility, *ranking query* (Guentzer, Balke, & Kiessling, 2000; Fagin, Lotem, & Naor, 2003) of returning few highly preferred results has been studied, while to maximize qualitative utility, *skyline query* (Börzsönyi, Kossmann, & Stocker, 2001; Godfrey, Shipley, & Gryz, 2007) of returning pareto-optimal objects not less preferred to (or "dominated" by) any other object based on the given

qualitative orderings, as we will discuss in detail in the personalization section.

Preference Mining

While user preference can be explicitly stated, in the form of a utility function or total orderings, such formulation can be too complicated for most end-users. Most applications thus adopt the approach of mining preferences from implicit feedbacks. In particular, preference mining from user click logs has been actively studied. Intuitively, items clicked by users can be considered as preferred items, over the items not clicked, which suggests *qualitative preference* of the specific user. More recently, the problem of using such qualitative reference information to infer *quantitative preference* utility function has been studied (Joachims, 2002; Radlinski & Joachims, 2005). These works adopt machine-learning approach to use qualitative orderings as training data to mine an underlying utility function.

Alternatively to *offline mining* of preferences from user logs, system may support dynamic and incremental preference elicitation procedures to collect additional user preference information and revise the result ranking. For *quantitative preference*, Yu, Hwang, and Chang (2005) studied adopting selective sampling to provide users with sample objects to provide feedbacks on, based on which the system collects information on user preferences and applies it in the retrieval process. To enable *online mining*, such sampling was designed to maximize the learning effectiveness such that the desired accuracy can be reached with the minimal user feedbacks. More recently, Joachims and Radlinski (2007) proposed to augment offline mining with online user elicitation procedure. For *qualitative preference*, Balke, Guentzer, and Lofi (2007) studied this online mining process to incrementally revise the skyline results, which was later extended to discuss a sophisticated user interface to assist users in the cooperative process of identifying partial orderings (Balke, Guentzer, & Lofi, 2007b).

Personalization

Once the preference is identified, we use it to retrieve the personalized results with respect to the preference.

For *quantitative preference*, the problem of efficient processing of ranking queries, which retrieve the results with the maximal utility score has been actively studied, pioneered by Algorithm FA (Fagin, 1996). Following works can be categorized into the two categories. First, more works followed to FA to be optimal in a stronger sense, by improving the stopping condition such that the upper bounds of the unseen objects can be more tightly computed, as presented in (Fagin et al., 2003). Second, another line of works follows to propose algorithms for various access scenarios, beyond FA assuming the sorted accesses over all predicates (Bruno, Gravano, & Marian, 2002; Hwang & Chang, 2005).

For *qualitative preference*, skyline queries are first studied as maximal vectors in (Kung, Luccio, & Preparata, 1975) and later adopted for data querying in (Börzsönyi et al., 2001) which proposes three basic skyline computation algorithms such as block nested loop (BNL), divide-and-conquer (D&C), and B-tree-based algorithms. Tan, Eng, and Ooi (2001) later study progressive skyline computation using auxiliary structures such as bitmap and sorted list. Kossmann, Ramsak, and Rost (2002) next propose nearest neighbor (NN) algorithm for efficiently pruning out dominated objects by iteratively partitioning the data space based on the nearest objects in the space. Meanwhile, Papadias, Tao, Fu, and Seeger (2003) develop branch and bound skyline (BBS) algorithm with I/O optimality property and Chomicki, Godfery, Gryz, and Liang (2003) develop sort-filter-skyline (SFS) algorithm leveraging pre-sorted lists for checking dominance condition efficiently. More recently, Godfrey, Shipley, and Gryz (2005) propose linear elimination-sort for skyline (LESS) algorithm with attractive average-case asymptotic time complexity, i.e., $O(d \cdot n)$ for d-dimensional data of size n.

More recently, there have been research efforts to combine the strength of these two query semantics. While skyline queries are highly intuitive to formulate, this intuitiveness comes with price of returning too many results especially when the dimensionality d of data is high, i.e., "curse of dimensionality" problem (Bentley, Kung, Schkolnick, & Thompson, 1978; Chaudhuri, Dalvi, & Kaushik, 2006; Godfrey, 2004) Recent efforts address this problem by narrowing down the skylines by ranking them and identifying the top-k results, which can be categorized into the following two groups of approaches: First, *user-oblivious ranking approach* leverages skyline frequency metric (Chan et al., 2006) which ranks each tuple in the decreasing order of the number of subspaces and in which the tuple is a skyline and k-dominances (Chan et al., 2006) which identifies k-dominant skylines as the common skyline

objects in all *k*-dimensional subspaces and ranks the tuples in the increasing order of *k*. Second, *personalized ranking approach* studies how to annotate skylines with different characteristics, as a basis for user-specific personalized ranking of skyline objects (Pei, Fu, Lin, & Wang, 2007; Pei, Jin, Ester, & Tao, 2005;Pei et al., 2006; Lee, You, & Hwang, 2007).

FUTURE TRENDS

While personalization has been actively studied when preferences are fully specified, more theories and techniques need to be developed in the near future for exploiting partial preferences. For instance, even when the preference of the specific user is only partially specified, a query on the preferences of prior users can retrieve the users with the similar taste, based on which we can complete the preference elicitation. For this purpose, scalable storage/query systems for user preferences is essential, which has not yet been actively studied, while such systems for data itself has been well studied.

CONCLUSION

As more and more data are getting accessible, it is getting more and more important to provide personalized results tailored for user-specific preference. Toward the goal, we identify preference modeling, mining, and personalization as major challenges: First, preference modeling enables to present user preference, collected from (1) offline mining of logs or (2) online mining of online user behaviors to guide the preference formulation. Once the preference is identified, the ultimate goal of this line of research is to enable effective and efficient personalization, to provide optimal results for the user-specific preference.

REFERENCES

Balke, W.-T., Guentzer, U., & Lofi, C. (2007a). Eliciting matters. controlling skyline sizes by incremental integration of user preferences. *In Proceeding of International Conference on Database Systems for Advanced Applications (DASFAA).*

Balke, W.-T., Guentzer, U., & Lofi, C. (2007b). User interaction support for incremental refinement of preference-based queries. *In Proceeding of Research Challenge in Information Science (RCIS).*

Bentley, J., Kung, H., Schkolnick, M., & Thompson, C. (1978). On the average number of maxima in a set of vectors and applications. *In Journal of the association for computing machinery.*

Börzsönyi, S., Kossmann, D., & Stocker, K. (2001). The skyline operator. *In Proceeding of International Conference on Data Engineering (ICDE).*

Bruno, N., Gravano, L., & Marian, A. (2002). Evaluating top-k queries over web-accessible databases. *In Proceeding of International Conference on Data Engineering (ICDE).*

Chan, C.-Y., Jagadish, H., Tan, K., Tung, A. K., & Zhang, Z. (2006). On high dimensional skylines. *In Proceeding of International Conference on Extending Database Technology (EDBT).*

Chan, C.-Y., Jagadish, H., Tan, K.-L., Tung, A. K., & Zhang, Z. (2006). Finding k-dominant skyline in high dimensional space. In Proceeding of ACM SIGMOD International Conference on Management of Data .

Chaudhuri, S., Dalvi, N., & Kaushik, R. (2006). Robust cardinality and cost estimation for skyline operator. *In Proceeding of International Conference on Data Engineering (ICDE).*

Chomicki, J., Godfery, P., Gryz, J., & Liang, D. (2003). Skyline with presorting. *In Proceeding of International Conference on Data Engineering (ICDE).*

Fagin, R. (1996). Combining fuzzy information from multiple systems. *In Proceeding of ACM SIGMOD Principles of Database Systems (PODS)*

Fagin, R., Lotem, A., & Naor, M. (2003). Optimal aggregation algorithms for middleware. *Journal of Computer and System Sciences*, 66 (4), 614-656.

Fishburn, P. C. (1970). Utility theory for decision making (No. 18). *Wiley*.

Godfrey, P. (2004). Skyline cardinality for relational processing. *In Foundations of information and knowledge systems.*

Godfrey, P., Shipley, R., & Gryz, J. (2005). Maximal vector computation in large data sets. *In Proceeding*

of International Conference on Very Large Database (VLDB).

Godfrey, P., Shipley, R., & Gryz, J. (2007). Algorithms and analyses for maximal vector computation. *The VLDB Journal, 16 (1), 5-28.*

Guentzer, U., Balke, W.-T., & Kiessling, W. (2000). Optimizing multi-feature queries for image databases. *In Proceeding of International Conference on Very Large Database (VLDB).*

Hwang, S., & Chang, K. C. (2005). Optimizing access cost for top-k queries over web sources. *In Proceeding of International Conference on Data Engineering (ICDE).*

Joachims, T. (2002). Optimizing search engines using clickthrough data. *In Proceeding of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.*

Joachims, T., & Radlinski, F. (2007). Active exploration for learning rankings from clickthrough data. *In Proceeding of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.*

Keeney, R. L., & Raiffa, H. (1976). Decisions with multiple objectives. preferences and value tradeoffs. *Wiley*.

Kossmann, D., Ramsak, F., & Rost, S. (2002). Shooting stars in the sky: An online algorithm for skyline queries. *In Proceeding of International Conference on Very Large Database (VLDB).*

Kung, H. T., Luccio, F., & Preparata, F. (1975). On finding the maxima of a set of vectors. *In Journal of the association for computing machinery.*

Lee, J., You, G., & Hwang, S. (2007). Telescope: Zooming to interesting skylines. *In Proceeding of International Conference on Database Systems for Advanced Applications (DASFAA).*

Lin, X., Yuan, Y., Zhang, Q., & Zhang, Y. (2007). Selecting stars: The k most representative skyline operator. *In Proceeding of International Conference on Data Engineering (ICDE).*

Papadias, D., Tao, Y., Fu, G., & Seeger, B. (2003). An optimal and progessive algorithm for skyline queries. *In Proceeding of ACM SIGMOD International Conference on Management of Data* .

Payne, J. W., Bettman, J. R., & Johnson, E. J. (1993). The adaptive decision maker. Cambridge University Press.

Pei, J., Fu, A. W., Lin, X., & Wang, H. (2007). Computing compressed multidimensional skyline cubes efficiently. *In Proceeding of International Conference on Data Engineering (ICDE).*

Pei, J., Jin, W., Ester, M., & Tao, Y. (2005). Catching the best views of skyline: A semantic approach based on decisive subspaces. *In Proceeding of International Conference on Very Large Database (VLDB).*

Pei, J., Yuan, Y., Lin, X., Jin, W., Ester, M., Liu, Q., et al. (2006). Towards multimimensional subspace skyline analysis. ACM *Transactions on Database Systems*.

Radlinski, F., & Joachims, T. (2005). Query chains: Learning to rank from implicit feedback. *In Proceeding of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.*

Tan, K., Eng, P., & Ooi, B. C. (2001). Efficient progressive skyline computation. *In Proceeding of International Conference on Very Large Database (VLDB).*

Yu, H., Hwang, S., & Chang, K. (2005). RankFP: A framework for supporting rank formulation and processing. In *Proceeding of International Conference on Data Engineering (ICDE).*

KEY TERMS

Online Preference Mining: Data mining process to identify preferences online from runtime user behaviors.

Personalized Retrieval: Procedures of tailoring retrieval to individual users' preferences.

Preference Elicitation: Procedures that support the expression of user-specific preferences, typically through the recommendations of data items.

Preference Mining: Data mining process to identify preferences from implicit user feedbacks.

Preference Modeling: Formal presentation of preferences in human and machine understandable forms.

Ranking Query: Query that supports the retrieval of data objects in the descending order of quantitative utility scores.

Skyline Query: Query that supports the retrieval of pareto-optimal results, not less preferred by any other object.

Privacy Preserving OLAP and OLAP Security

Alfredo Cuzzocrea
University of Calabria, Italy

Vincenzo Russo
University of Calabria, Italy

INTRODUCTION

The problem of ensuring the *privacy* and *security* of OLAP data cubes (Gray et al., 1997) arises in several fields ranging from advanced *Data Warehousing* (DW) and *Business Intelligence* (BI) systems to sophisticated *Data Mining* (DM) tools. In DW and BI systems, decision making analysts aim at avoiding that malicious users access perceptive ranges of multidimensional data in order to infer *sensitive knowledge*, or *attack* corporate data cubes via violating user rules, grants and revokes. In DM tools, domain experts aim at avoiding that malicious users infer *critical-for-the-task knowledge* from authoritative DM results such as frequent item sets, patterns and regularities, clusters, and discovered association rules. In more detail, the former application scenario (i.e., DW and BI systems) deals with both the privacy preservation and the security of data cubes, whereas the latter one (i.e., DM tools) deals with *privacy preserving OLAP issues* solely. With respect to security issues, although security aspects of information systems include a plethora of topics ranging from *cryptography* to *access control* and *secure digital signature*, in our work we particularly focus on *access control techniques* for data cubes, and remand the reader to the active literature for the other orthogonal matters.

Specifically, privacy preservation of data cubes refers to the problem of ensuring the privacy of data cube cells (and, in turn, that of queries defined over collections of data cube cells), i.e. hiding sensitive information and knowledge during data management activities, according to the general guidelines drawn by Sweeney in her seminar paper (Sweeney, 2002), whereas access control issues refer to the problem of ensuring the security of data cube cells, i.e. restricting the access of unauthorized users to specific sub-domains of the target data cube, according to well-known concepts studied and assessed in the context of DBMS security. Nonetheless, it is quite straightforward foreseeing that these two even distinct aspects should be meaningfully *integrated* in order to ensure both the privacy and security of complex data cubes, i.e. data cubes built on top of complex data/knowledge bases.

During last years, these topics have became of great interest for the Data Warehousing and Databases research communities, due to their exciting theoretical challenges as well as their relevance and practical impact in modern real-life OLAP systems and applications. On a more conceptual plane, theoretical aspects are mainly devoted to study how *probability* and *statistics schemes* as well as rule-based models can be applied in order to efficiently solve the above-introduced problems. On a more practical plane, researchers and practitioners aim at integrating convenient privacy preserving and security solutions within the core layers of commercial OLAP server platforms.

Basically, to tackle deriving privacy preservation challenges in OLAP, researchers have proposed models and algorithms that can be roughly classified within two main classes: *restriction-based techniques*, and *data perturbation techniques*. First ones propose limiting the number of query kinds that can be posed against the target OLAP server. Second ones propose perturbing data cells by means of random noise at various levels, ranging from schemas to queries. On the other hand, access control solutions in OLAP are mainly inspired by the wide literature developed in the context of controlling accesses to DBMS, and try to adapt such schemes in order to control accesses to OLAP systems.

Starting from these considerations, in this article we propose a survey of models, issues and techniques in a broad context encompassing privacy preserving and security aspects of OLAP data cubes.

BACKGROUND

Handling sensitive data, which falls in privacy preserving issues, is common in many real-life application scenarios. For instance, consider a government agency that collects information about client applications/users for a specific *e*-government process/task, and then makes this information available for a third-party agency willing to perform market analysis for business purposes. In this case, preserving sensitive data of client applications/users and protecting their utilization from malicious behaviors play a leading role. It should be taken into account that this scenario gets worse in OLAP systems, as the interactive nature of such systems *naturally* encourages malicious users to retrieve *sensitive knowledge* by means of *inference techniques* (Wang et al., 2004a; Wang et al., 2004b) that, thanks to the wide availability of OLAP tools and operators (Han & Kamber, 2000), can reach an high degree of effectiveness and efficiency.

Theoretical background of privacy preserving issues in OLAP relies on research experiences in the context of *statistical databases* (Shoshani, 1997), where these issues have been firstly studied. In statistical databases, this problem has been tackled by means of *Statistical Disclosure Control* (SDC) techniques (Domingo-Ferrer, 2002), which propose achieving the privacy preservation of data via *trade-offing the accuracy and privacy of data*. The main idea of such an approach is that of admitting the need for data provisioning while, at the same time, the need for privacy of data. In fact, *full data hiding* or *full data camouflaging* are both useless, as well as publishing *completely-disclosed data sets*. Therefore, balancing accuracy and privacy of data is a reasonable solution to this challenge. In this context, two meaningful measures for evaluating the accuracy and privacy preservation capabilities of an arbitrary method/technique have been introduced. The first one is referred as *Information Loss* (IL). It allows us to estimate the lost of information (i.e., the accuracy decrease) due to a given privacy preserving method/technique. The second one is the *Disclosure Risk* (DR). It allows us to estimate the risk of disclosing sensitive data due to a given privacy preserving method/technique.

Duncan *et al.* (2001) introduce two metrics for probabilistically evaluating IL and DR. Given a numerical attribute A that can assume a value w with probability P_w, such that $\mathcal{D}(w)$ is the domain of w (i.e., the set of *all* the values that A can assume), a possible metrics of IL is given by the *Data Utility* (DU), which is defined as follows:

$$DU(w) = \frac{|\mathcal{D}(w)|}{\sum_{\mathrm{w}} P_w \cdot (w-1)^2} \tag{1}$$

where $|\mathcal{D}(w)|$ denotes the cardinality of $\mathcal{D}(w)$. It should be noted that DU and IL are inversely proportional, i.e. the more is IL the less is DU, and, conversely, the less is IL the more is DU.

Different formulations exist. For instance, Sung *et al.* (2006) introduce the so-called *accuracy factor* $F_{a,Q}$ of a given query Q against a data cube D, i.e. the relative accuracy decrease of the *approximate answer* to Q, denoted by $\tilde{A}(Q)$, which is evaluated on the *synopsis data cube* $\tilde{D}$ obtained from D by means of *perturbation-based techniques* (presented next), with respect to the *exact answer* to Q, denoted by $A(Q)$, which is evaluated on the original data cube D. $F_{a,Q}$ is defined as follows:

$$F_{a,Q} = 2^{-\left|\frac{\tilde{A}(Q)-A(Q)}{A(Q)}\right|} \tag{2}$$

With regards to DR, Duncan *et al.* (2001) consider the *probability with respect to the malicious user* that A can assume the value w, denoted by P_w^U, and introduce the following metrics that models DR in terms of the reciprocal of the *information entropy*, as follows:

$$DR(w) = \frac{1}{-\sum_{\mathrm{w}} P_w^U \cdot \log(P_w^U)} \tag{3}$$

Indeed, being impossible to estimate the value of P_w^U, as one should know *all* the information/knowledge held by the malicious user, in (Duncan et al., 2001) the *conditional* version of (3) is proposed as follows:

$$DR(w) = \frac{1}{-\sum_{\mathrm{w}} p(w \mid u) \cdot \log_2 p(w \mid u)} \tag{4}$$

such that $p(w|u)$ denotes the *conditional probability* that the actual value of A is w while the value known by the malicious user is u.

Just like for IL, different formulations for measuring DR exist. Sung *et al.* (2006) introduce the so-called *privacy factor* $F_{p,D}$ of a given data cube D with respect to the corresponding perturbation-based synopsis data cube $\tilde{D}$. $F_{p,D}$ is defined as follows:

$$F_{p,D} = \frac{1}{N} \cdot \sum_{i=1}^{N} \frac{\left|\tilde{C}_i - C_i\right|}{\left|C_i\right|} \quad (5)$$

such that C_i denotes a data cell of the data cube D, and $\tilde{C}$ the corresponding perturbed data cell $\tilde{C}$ of the synopsis data cube $\tilde{D}$.

According to research results presented in (Duncan et al., 2001), accuracy and privacy of a privacy preserving technique are related and must be traded-off. Intuitively enough, an increase of one of these properties causes a correlated decrease of the other one. Also, it is a matter to notice that having maximum accuracy implies a very high DR while, conversely, minimum accuracy implies minimum DR. On the other hand, accuracy cannot be minimized, and DR cannot be maximized. As we will describe in next section, most of privacy preserving OLAP techniques of the active literature are based on this terminology, and on the fundamental idea of trading-off accuracy and privacy.

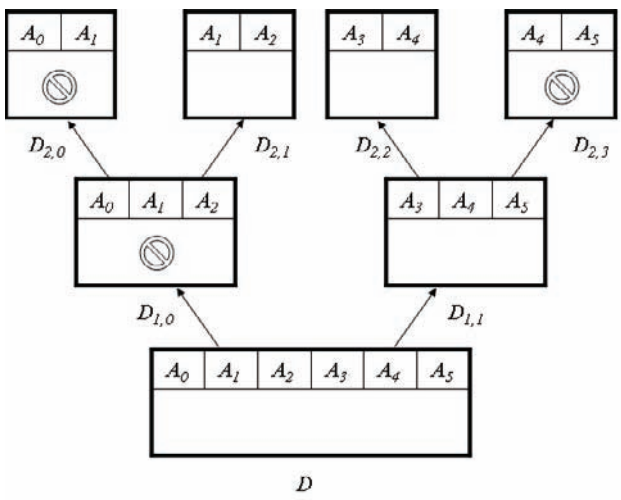
For what regards the background of security issues, actual literature focuses on access control techniques, as discussed above. Access control has a quite long history in DBMS, where the goal is protecting data objects from unauthorized accesses. In this context, an *authorization* is modeled as a triple: ⟨*Object*, *Subject*, +/-*Action*⟩, such that (*i*) *Object* is the data object to be protected, (*ii*) *Subject* is the user/application accessing the object *Object*, and (*iii*) *Action* is the operation that the user/application *Subject* can or cannot (i.e., +/-) perform on the object *Object*. Typically, read-only operations are controlled, since data updates are indeed allowed to few users/user-groups only.

Because of (very) different data models, the main difference between access control issues in DBMS and OLAP systems is represented by the nature of data objects. In DBMS, an object can be a data table, a query or a record. This allows us to achieve very precise authorization mechanisms, as singleton records or partitions of data tables (they may be vertical or horizontal) can be handled. Indeed, we can claim that access control techniques take great advantages from the *flexibility* of DBMS models and schemas. When OLAP systems are considered, models and schemas are characterized by *dimensionality and multi-resolution of data*, and access control mechanisms need to become more sophisticated in order to prevent malicious user attacks. Target data objects are represented by data cubes, which can be generally thought as collections of *cuboids*, one for each combination of hierarchy levels. Also, dimensional hierarchies pose additional challenges, as malicious users can successfully exploit the multi-resolution data models defined by dimensional hierarchies in order to devise very effective *knowledge inference techniques* able to hierarchically browse the structure of the cube with the goal of discovering aggregations computed over sensitive ranges of data. Also, rich OLAP tools and operators (Han & Kamber, 2000), such as *roll-up*, *drill-down* and *slice & dice*, represent "sophisticate instruments" in the hands of malicious users, as they can be used to extract sensitive knowledge from a secured cuboid at a given level ℓ starting from disclosed cuboids at levels different from ℓ.

To overcome security breaches like those described above, access control techniques for OLAP data cubes usually apply a restriction-based approach, and limit the set of cuboids that can be accessed by external applications/users. Nevertheless, even in this case a trade-off strategy must be devised, as securing a large number of cuboids can become useless in real-life OLAP scenarios. All considering, as studied by Wang *et al.* in (Wang et al., 2004a; Wang et al., 2004b), the *inference problem* in OLAP introduces more probing issues rather than precursor scientific areas related to inference issues in statistical databases (Denning & Schlorer, 1983).

To give an example on a simple access control mechanism in *Relational OLAP* (ROLAP) data cubes (i.e., data cubes stored in form of tables of a RDBMS in Han & Kamber, 2000), consider Figure 1, where a

Figure 1. Access control mechanism on a ROLAP data cube



data cube D with the related *cuboid lattice* is depicted. In this case, the cuboids $D_{1,0}$, $D_{2,0}$, and $D_{2,3}$ are secured to authorized applications/users only and forbidden to unauthorized ones.

PRIVACY PRESERVING TECHNIQUES IN OLAP

Privacy Preserving OLAP (PPOLAP) (Agrawal et al., 2005) is a specialized case of *Privacy Preserving Data Mining* (PPDM) (Agrawal et al., 2000). While PPDM concerns with the privacy of data during DM activities (e.g., clustering, classification, pattern discovery, association rule discovery etc), PPOLAP deals with the problem of preserving the privacy of data cells of a given data cube during typical OLAP activities such as performing classical operators (e.g., roll-up and drill-down) or evaluating complex OLAP queries (e.g., *range-* (Ho et al., 1997), *top-k* (Xin et al., 2006), and *iceberg* (Fang et al., 1998) queries). With respect to PPDM, PPOLAP introduces more *semantics* into the privacy preservation due to its well-known knowledge-intensive tools such as multidimensionality and multi-resolution of data, and hierarchies.

In the following, we review the two kinds of privacy preserving techniques in OLAP (i.e., restriction-based and perturbation-based techniques) introduced in the previous Section.

Restriction-based techniques limit the queries that can be posed to the OLAP server in order to preserve the privacy of data cells. This problem is related to the issue of *auditing queries in statistical databases*, which consists in analyzing the past (answered) queries in order to determine whether these answers can be composed by a malicious user to infer sensitive knowledge in the form of answers to forbidden queries. Therefore, in order to understand which kinds of queries must be forbidden, a restriction-based technique needs to audit queries posed to the target data (e.g., OLAP) server during a given interval of time. Auditing queries in statistical databases is the conceptual and theoretical basis of auditing queries in OLAP systems.

Interesting auditing techniques for queries against statistical databases have been proposed by Dobkin *et al.* (1979), which introduce a model for auditing average and median queries, and Chin and Ozsoyoglu (1982), which propose a technique for handling the past history of SUM queries in order to reduce the sequence of answered queries to privacy preservation purposes. Also, Chin and Ozsoyoglu (1982) describe how to check the *compromisability* of the underlying statistical database when using the reduced sequence. The proposed auditing technique is called *Audit Expert*.

More recently, few approaches focusing on the problem of auditing techniques for OLAP data cubes and queries appeared. Among all, we recall: (*i*) the work of Zhang *et al.* (2004), which propose an interesting *information theoretic approach* that simply counts the number of cells already covered to answer previous queries in order to establish if a *new* query should be answered or not; (*ii*) the work of Malvestuto *et al.* (2006), which introduce a novel notation for auditing range-SUM queries (i.e., an OLAP-like class of queries) against statistical databases making use of *Integer Linear Programming* (ILP) tools for detecting if a *new* range-sum query can be answered safely.

Perturbation-based techniques add random noise at various levels of the target database, ranging from schemas, like in (Schlorer, 1981), to query answers, like in (Beck, 1980).

Agrawal *et al.* (2005) first propose the notion of PPOLAP. They define a PPOLAP model over data partitioned across multiple clients using a *randomization approach* on the basis of which (*i*) clients perturb tuples which with they participate to the partition in order to gain *row-level privacy*, and (*ii*) server is capable of evaluating OLAP queries against perturbed tables via reconstructing original distributions of attributes involved by such queries. In (Agrawal et al., 2005), authors demonstrate that the proposed approach is safe against privacy breaches.

Hua *et al.* (2005) propose a different approach to preserve the privacy of OLAP data cubes. They argue that hiding parts of data that could cause inference of sensitive cuboids is enough in order to achieve the notion of "secure" data cubes. While a strengthness point of the proposed approach is represented by its simplicity, authors do not provide sufficient experimental analysis to prove in which measure the data hiding phase affects the target OLAP server.

Sung *et al.* (2006) propose a *random data distortion* technique, called *zero-sum method*, for preserving secret information of *individual* data cells while providing accurate answers to range-queries over original aggregates. Roughly speaking, data distortion consists in iteratively altering the values of individual data cells of the target data cube in such a way as to maintain the

marginal sums of data cells along rows and columns of the data cube equal to zero. This ensures the privacy of individual data cells, and the correctness of answers to range-queries.

Due to different, specific motivations, both restriction-based and perturbation-based techniques are ineffective in OLAP. Specifically, restriction-based techniques cannot be applied to OLAP systems since the nature of such systems is intrinsically *interactive*, and based on a wide set of operators and query classes. On the other hand, perturbation-based techniques cannot be applied in OLAP systems since they introduce excessive computational overheads when executed on massive data cubes.

SECURITY TECHNIQUES IN OLAP

The problem of security control methods has been widely studied in the context of statistical databases, and it has produced a wide and consolidate literature (Adam & J.C. Wortmann, 1989) that, in turn, is inspiring actual initiatives for securing OLAP data cubes, i.e. limiting their access to authorized applications/users only.

As stated in the background Section, the main idea of these approaches is devising *access control schemes* that establish how applications/users must access multidimensional data on the basis of *grants* and *revokes* (Griffiths & Wade, 1976), *roles* (Sandhu et al., 1996), and *authorization rules* (Jajodia et al., 2001).

Following the above-mentioned pioneristic approaches, some preliminary, sporadic studies in the context of securing data warehouses (Bhargava, 2000) and data cubes (Pernul & Priebe, 2000) have been appeared in literature subsequently. While these works are clearly in their initial stages, they have inspired most part of actual research effort in the context of access control schemes for OLAP data cubes.

(Wang et al., 2004a; Wang et al., 2004b) represent the state-of-the-art for access control schemes in OLAP. They propose a novel technique for limiting inference breaches in OLAP systems via detecting *cardinality-based* sufficient conditions over cuboids, in order to make data cubes safe with respect to malicious users. Specifically, the proposed technique combines access control and inference control techniques (Denning & Schlorer, 1983), being (*i*) first one based on the hierarchical nature of data cubes in terms of cuboid lattice and multi-resolution of data, and (*ii*) second one based on directly applying restriction to *coarser aggregations* of data cubes, and then removing remaining inferences that can be still derived.

FUTURE TRENDS

Privacy preserving and security issues in OLAP will play more and more a significant role in Data Warehousing research. Actually, it seems that the following topics will decisively capture the attention from the research communities:

- *Low-Complexity Solutions*: since OLAP data cubes are massive in size, the problem of devising efficient solutions requiring low spatio-temporal complexity is a challenge of great interest.
- *Enhancing the Semantics of Privacy Preserving Data Cubes*: semantics of data cube models is richer and attracting than semantics of database models; this offers noteworthy extensions of the actual privacy preserving schemes towards the integration of such schemes with innovative paradigms such as *flexibility* and *adaptivity*.
- *Scaling-Up Access Control Models*: with regards to access control issues, devising models able to scale-up on complex data cubes (i.e., data cubes with complex dimensions, hierarchies and measures) represents the next work to be accomplished.
- *Integration with Real-Life Data Warehouse Servers*: after their complete assessment, both privacy preserving and security models need to be integrated and put-at-work within the core layer of real-life Data Warehouse servers, in order to precisely evaluate their effectiveness in realistic settings and operational environments.

CONCLUSION

Inspired by privacy preserving and security issues in statistical databases, privacy preserving and secure OLAP data cubes are a leading topic in Data Warehousing research. Despite this, researchers have not devoted a great attention to this area so far. As a consequence, privacy preservation and security of data cubes is a relatively adolescent research context, which needs to be further developed during next years. Under such

a futuristic vision, this article has presented a survey of state-of-the-art models, issues and techniques in a broad context encompassing privacy preserving and security aspects of OLAP data cubes, with critical analysis of benefits and limitations coming from these initiatives.

REFERENCES

Adam, N.R., & Wortmann, J.C. (1989). Security-Control Methods for Statistical Databases: A Comparative Study. *ACM Computing Surveys*, 21(4), 515-556.

Agrawal, R., & Srikant, R. (2000). Privacy-Preserving Data Mining. *Proceedings of the 2000 ACM International Conference on Management of Data*, 439-450.

Agrawal, R., Srikant, R., & Thomas, D. (2005). Privacy-Preserving OLAP. *Proceedings of the 2005 ACM International Conference on Management of Data*, 251-262.

Beck, L.L. (1980). A Security Mechanism for Statistical Databases. *ACM Transactions on Database Systems*, 5(3), 316-338.

Bhargava, B.K. (2000). Security in Data Warehousing. *Proceedings of the 2nd International Conference on Data Warehousing and Knowledge Discovery*, 287-289.

Chin, F.Y., & Ozsoyoglu, G. (1982). Auditing and Inference Control in Statistical Databases. *IEEE Transactions on Software Engineering*, 8(6), 574-582.

Denning, D.E., & Schlorer, J. (1983). Inference Controls for Statistical Databases. *IEEE Computer*, 16(7), 69-82.

Dobkin, D., Jones, A.K., & Lipton, R.J. (1979). Secure Databases: Protection against User Influence. *ACM Transactions on Database Systems*, 4(1), 97-106.

Domingo-Ferrer, J. (ed.) (2002). *Inference Control in Statistical Databases: From Theory to Practice*, Springer.

Duncan, G.T., Fienberg, S.E., Krishnan, R., Padman, R., & Roehrig, S.F. (2001). Disclosure Limitation Methods and Information Loss for Tabular Data. *P. Doyle, J. Lane, J. Theeuwes, and L. Zayatz (eds.), "Confidentiality, Disclosure and Data Access: Theory and Practical Applications for Statistical Agencies"*, Elsevier, 135-166.

Fang, M., Shivakumar, N., Garcia-Molina, H., Motwani, R., & Ullman, J.D. (1998). Computing Iceberg Queries Efficiently. *Proceedings of the 24th International Conference on Very Large Data Bases*, 299-310.

Gray, J., Chaudhuri, S., Bosworth, A., Layman, A., Reichart, D., & Venkatrao, M. (1997). Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals. *Data Mining and Knowledge Discovery*, 1(1), 29-53.

Griffiths, P., & Wade, B.W. (1976). An Authorization Mechanism for a Relational Database System. *ACM Transactions on Database Systems*, 1(3), 242-255.

Han, J., & Kamber, M. (2000). *Data Mining: Concepts and Techniques*. Morgan Kauffmann Publishers.

Ho, C.-T., Agrawal, R., Megiddo, N., & Srikant, R. (1997). Range Queries in OLAP Data Cubes. *Proceedings of the 1997 ACM International Conference on Management of Data*, 73-88.

Jajodia, S., Samarati, P., Sapino, M.L., & Subrahmanian, V.S. (2001). Flexible Support for Multiple Access Control Policies. *ACM Transactions on Database Systems*, 26(4), 1-57.

Malvestuto, F.M., Mezzani, M., & Moscarini, M. (2006). Auditing Sum-Queries to Make a Statistical Database Secure. *ACM Transactions on Information and System Security*, 9(1), 31-60.

Pernul, G., & Priebe, T. (2000). Towards OLAP Security Design – Survey and Research Issues. *Proceedings of the 3rd ACM International Workshop on Data Warehousing and OLAP*, 114-121.

Sandhu, R.S., Coyne, E.J., Feinstein, H.L., & Youman, C.E. (1996). Role-based Access Control Models. *IEEE Computer*, 29(2), 38-47.

Schlorer, J. (1981). Security of Statistical Databases: Multidimensional Transformation. *ACM Transactions on Database Systems*, 6(1), 95-112.

Shoshani, A (1997). OLAP and Statistical Databases: Similarities and Differences. *Proceedings of the 16th ACM International Symposium on Principles of Database Systems*, 185-196.

Sung, S.Y., Liu, Y., Xiong, H., & Ng, P.A. (2006). Privacy Preservation for Data Cubes. *Knowledge and Information Systems*, 9(1), 38-61.

Sweeney, L. (2002). *k*-Anonymity: A Model for Protecting Privacy. *International Journal on Uncertainty Fuzziness and Knowledge-based Systems*, 10(5), 557-570.

Wang, L., Jajodia, S., & Wijesekera, D. (2004). Securing OLAP Data Cubes against Privacy Breaches. *Proceedings of the 2004 IEEE Symposium on Security and Privacy*, 161-175.

Wang, L., Jajodia, S., & Wijesekera, D. (2004). Cardinality-based Inference Control in Data Cubes. *Journal of Computer Security*, 12(5), 655-692.

Zhang, N., Zhao, W., & Chen, J. (2004). Cardinality-based Inference Control in OLAP Systems: An Information Theoretic Approach. *Proceedings of the 7th ACM International Workshop on Data Warehousing and OLAP*, 59-64.

Xin, D., Han, J., Cheng, H., & Li, X. (2006). Answering Top-k Queries with Multi-Dimensional Selections: The Ranking Cube Approach. *Proceedings of the 32nd International Conference on Very Large Data Bases*, 463-475.

KEY TERMS

Access Control Scheme: A scheme that establishes the constraints under which a data object must be accessed.

Disclosure Risk: A measure of the amount of disclosed information due to a privacy preserving technique.

Inference: A methodology for extracting sensitive knowledge from disclosed information.

Information Loss: A measure of the information lost due to a privacy preserving technique.

On-Line Analytical Processing (OLAP): A methodology for representing, managing and querying massive DW data according to multidimensional and multi-resolution abstractions of them.

Privacy Preservation: A property that describes the capability of data management techniques in preserving the privacy of processed data.

Security: A property that describes the capability of data representation models in ensuring the security of portions of data whose access is forbidden to unauthorized applications/users.

Privacy-Preserving Data Mining

Stanley R. M. Oliveira
Embrapa Informática Agropecuária, Brazil

INTRODUCTION

Despite its benefits in various areas (e.g., business, medical analysis, scientific data analysis, etc), the use of data mining techniques can also result in new threats to privacy and information security. The problem is not data mining itself, but the way data mining is done. "Data mining results rarely violate privacy, as they generally reveal high-level knowledge rather than disclosing instances of data" (Vaidya & Clifton, 2003). However, the concern among privacy advocates is well founded, as bringing data together to support data mining projects makes misuse easier. Thus, in the absence of adequate safeguards, the use of data mining can jeopardize the privacy and autonomy of individuals.

Privacy-preserving data mining (PPDM) cannot simply be addressed by restricting data collection or even by restricting the secondary use of information technology (Brankovic & V. Estivill-Castro, 1999). Moreover, there is no exact solution that resolves privacy preservation in data mining. In some applications, solutions for PPDM problems might meet privacy requirements and provide valid data mining results (Oliveira & Zaïane, 2004b).

We have witnessed three major landmarks that characterize the progress and success of this new research area: *the conceptive landmark*, *the deployment landmark*, and *the prospective landmark*. *The Conceptive landmark* characterizes the period in which central figures in the community, such as O'Leary (1995), Piatetsky-Shapiro (1995), and others (Klösgen, 1995; Clifton & Marks, 1996), investigated the success of knowledge discovery and some of the important areas where it can conflict with privacy concerns. The key finding was that knowledge discovery can open new threats to informational privacy and information security if not done or used properly.

The Deployment landmark is the current period in which an increasing number of PPDM techniques have been developed and have been published in refereed conferences. The information available today is spread over countless papers and conference proceedings. The results achieved in the last years are promising and suggest that PPDM will achieve the goals that have been set for it.

The Prospective landmark is a new period in which directed efforts toward standardization occur. At this stage, there is no consensus on privacy principles, policies, and requirements as a foundation for the development and deployment of new PPDM techniques. The excessive number of techniques is leading to confusion among developers, practitioners, and others interested in this technology. One of the most important challenges in PPDM now is to establish the groundwork for further research and development in this area.

BACKGROUND

Understanding privacy in data mining requires understanding how privacy can be violated and the possible means for preventing privacy violation. In general, one major factor contributes to privacy violation in data mining: the misuse of data.

Users' privacy can be violated in different ways and with different intentions. Although data mining can be extremely valuable in many applications (e.g., business, medical analysis, etc), it can also, in the absence of adequate safeguards, violate informational privacy. Privacy can be violated if personal data are used for other purposes subsequent to the original transaction between an individual and an organization when the information was collected (Culnan, 1993).

Defining Privacy for Data Mining

In general, privacy preservation occurs in two major dimensions: users' personal information and information concerning their collective activity. We refer to the former as individual privacy preservation and the latter

as collective privacy preservation, which is related to corporate privacy in (Clifton et al., 2002).

- **Individual privacy preservation:** The primary goal of data privacy is the protection of personally identifiable information. In general, information is considered personally identifiable if it can be linked, directly or indirectly, to an individual person. Thus, when personal data are subjected to mining, the attribute values associated with individuals are private and must be protected from disclosure. Miners are then able to learn from global models rather than from the characteristics of a particular individual.
- **Collective privacy preservation:** Protecting personal data may not be enough. Sometimes, we may need to protect against learning sensitive knowledge representing the activities of a group. We refer to the protection of sensitive knowledge as collective privacy preservation. The goal here is quite similar to that one for statistical databases, in which security control mechanisms provide aggregate information about groups (population) and, at the same time, prevent disclosure of confidential information about individuals. However, unlike as is the case for statistical databases, another objective of collective privacy preservation is to protect sensitive knowledge that can provide competitive advantage in the business world.

MAIN FOCUS

A Taxonomy of existing PPDM Techniques

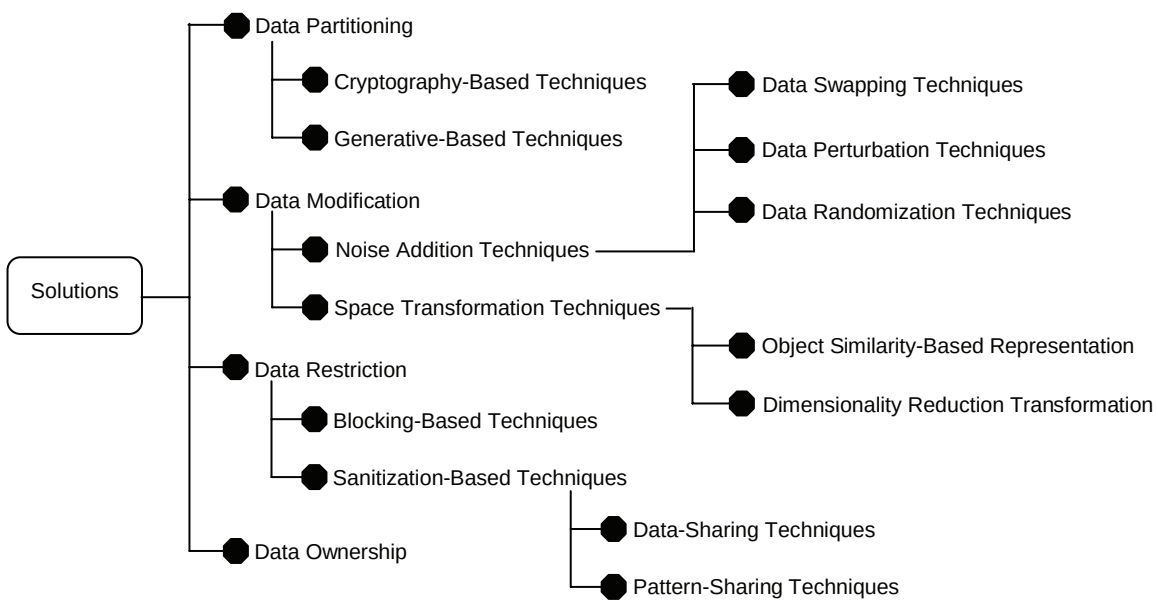
The existing PPDM techniques in the literature can be classified into four major categories: data partitioning, data modification, data restriction, and data ownership as can be seen in Figure 1.

Data Partitioning Techniques

Data partitioning techniques have been applied to some scenarios in which the databases available for mining are distributed across a number of sites, with each site only willing to share data mining results, not the source data. In these cases, the data are distributed either horizontally or vertically. In a horizontal partition, different entities are described with the same schema in all partitions, while in a vertical partition the attributes of the same entities are split across the partitions. The existing solutions can be classified into Cryptography-Based Techniques and Generative-Based Techniques.

- **Cryptography-based techniques:** In the context of PPDM over distributed data, cryptography-based techniques have been developed to solve problem of the following nature: two or more

Figure 1. A taxonomy of PPDM techniques.



parties want to conduct a computation based on their private inputs. The issue here is how to conduct such a computation so that no party knows anything except its own input and the results. This problem is referred to as the secure multiparty computation (SMC) problem (Goldreich, Micali, & Wigderson, 1987). The technique proposed in (Lindell & Pinkas, 2000) address privacy-preserving classification, while the techniques proposed in (Kantarcioğlu & Clifton, 2002; Vaidya & Clifton, 2002) address privacy-preserving association rule mining, and the technique in (Vaidya & Clifton, 2003) addresses privacy-preserving clustering.

- **Generative-based techniques:** These techniques are designed to perform distributed mining tasks. In this approach, each party shares just a small portion of its local model which is used to construct the global model. The existing solutions are built over horizontally partitioned data. The solution presented in (Veloso et al., 2003) addresses privacy-preserving frequent itemsets in distributed databases, whereas the solution in (Meregu & Ghosh, 2003) addresses privacy-preserving distributed clustering using generative models.

Data Modification Techniques

Data modification techniques modify the original values of a database that needs to be shared, and in doing so, privacy preservation is ensured. The transformed database is made available for mining and must meet privacy requirements without losing the benefit of mining. In general, data modification techniques aim at finding an appropriate balance between privacy preservation and knowledge disclosure. Methods for data modification include noise addition techniques and space transformation techniques.

- **Noise addition techniques:** The idea behind noise addition techniques for PPDM is that some noise (e.g., information not present in a particular tuple or transaction) is added to the original data to prevent the identification of confidential information relating to a particular individual. In other cases, noise is added to confidential attributes by randomly shuffling the attribute values to prevent the discovery of some patterns that are not supposed to be discovered. We categorize noise addition techniques into three groups: (1) data swapping techniques that interchange the values of individual records in a database (Estivill-Castro & Brankovic, 1999); (2) data distortion techniques that perturb the data to preserve privacy, and the distorted data maintain the general distribution of the original data (Liu, Kargupta, & Ryan, 2006; Agrawal & Srikant, 2000); and (3) data randomization techniques which allow one to perform the discovery of general patterns in a database with error bound, while protecting individual values. Like data swapping and data distortion techniques, randomization techniques are designed to find a good compromise between privacy protection and knowledge discovery (Evfimievski et al., 2002; Rizvi & Haritsa, 2002; Zang, Wang, & Zhao, 2004).
- **Space transformation techniques:** These techniques are specifically designed to address privacy-preserving clustering. These techniques are designed to protect the underlying data values subjected to clustering without jeopardizing the similarity between objects under analysis. Thus, a space transformation technique must not only meet privacy requirements but also guarantee valid clustering results. The solution proposed in (Oliveira & Zaïane, 2007a; Oliveira & Zaïane, 2007b) relies on dimensionality reduction-based transformation. The idea behind such a solution is that by reducing the dimensionality of a dataset to a sufficiently small value, one can find a trade-off between privacy, communication cost, and accuracy. Once the dimensionality of a database is reduced, the released database preserves (or slightly modifies) the distances between data points.

Data Restriction Techniques

Data restriction techniques focus on limiting the access to mining results through either generalization or suppression of information (e.g., items in transactions, attributes in relations), or even by blocking the access to some patterns that are not supposed to be discovered. Such techniques can be divided into two groups: Blocking-based techniques and Sanitization-based techniques.

- **Blocking-based techniques:** These techniques aim at hiding some sensitive information when

data are shared for mining. The private information includes sensitive association rules and classification rules that must remain private. Before releasing the data for mining, data owners must consider how much information can be inferred or calculated from large databases, and must look for ways to minimize the leakage of such information. In general, blocking-based techniques are feasible to recover patterns less frequent than originally since sensitive information is either suppressed or replaced with unknowns to preserve privacy (Saygin, Verykios, & Clifton, 2001).

- **Sanitization-based techniques:** Unlike blocking-based techniques that hide sensitive information by replacing some items or attribute values with unknowns, sanitization-based techniques hide sensitive information by strategically suppressing some items in transactional databases, or even by generalizing information to preserve privacy in classification. These techniques can be categorized into two major groups: (1) data-sharing techniques in which the sanitization process acts on the data to remove or hide the group of sensitive association rules that contain sensitive knowledge. To do so, a small number of transactions that contain the sensitive rules have to be modified by deleting one or more items from them or even adding some noise, i.e., new items not originally present in such transactions (Verykios et al., 2004; Oliveira & Zaïane, 2006); and (2) pattern-sharing techniques in which the sanitizing algorithm acts on the rules mined from a database, instead of the data itself. The existing solution removes all sensitive rules before the sharing process and blocks some inference channels (Oliveira, Zaïane, & Saygin, 2004a).

Data Ownership Techniques

Data ownership techniques can be applied to two different scenarios: (1) to protect the ownership of data by people about whom the data were collected (Felty & Matwin, 2002). The idea behind this approach is that a data owner may prevent the data from being used for some purposes and allow them to be used for other purposes. To accomplish that, this solution is based on encoding permissions on the use of data as theorems about programs that process and mine the data. Theorem proving techniques are then used to guarantee that these programs comply with the permissions; and (2) to identify the entity that receives confidential data when such data are shared or exchanged (Mucsi-Nagy & Matwin, 2004). When sharing or exchanging confidential data, this approach ensures that no one can read confidential data except the receiver(s). It can be used in different scenarios, such as statistical or research purposes, data mining, and on-line business-to-business (B2B) interactions.

FUTURE RESEARCH TRENDS

Preserving privacy on the Web has an important impact on many Web activities and Web applications. In particular, privacy issues have attracted a lot of attention due to the growth of e-commerce and e-business. These issues are further complicated by the global and self-regulatory nature of the Web.

Privacy issues on the Web are based on the fact that most users want to maintain strict anonymity on Web applications and activities. The ease access to information on the Web coupled with the ready availability of personal data, also made it easier and more tempting for interested parties (e.g., businesses and governments) to willingly or inadvertently intrude on individuals' privacy in unprecedented ways.

Clearly, privacy issues on Web data is an umbrella that encompasses many Web applications such as e-commerce, stream data mining, multimedia mining, among others. In this work, we focus on issues toward foundation for further research in PPDM on the Web because these issues will certainly play a significant role in the future of this new area. In particular, a common framework for PPDM should be conceived, notably in terms of definitions, principles, policies, and requirements. The advantages of a framework of that nature are as follows: (a) a common framework will avoid confusing developers, practitioners, and many others interested in PPDM on the Web; (b) adoption of a common framework will inhibit inconsistent efforts in different ways, and will enable vendors and developers to make solid advances in the future of research in PPDM on the Web.

CONCLUSION

The area of data mining has received special attention since the 1990s. The fascination with the promise of analysis of large volumes of data has led to an increasing number of successful applications of data mining in recent years. Despite its benefits in various areas, the use of data mining techniques can also result in new threats to privacy and information security. The problem is not data mining itself, but the way data mining is done. Thus, in the absence of adequate safeguards, the use of data mining can jeopardize the privacy and autonomy of individuals.

This chapter emphasizes issues concerning privacy-preserving data mining (PPDM), which has posed challenges for novel uses of data mining technology. These technical challenges indicate a pressing need to rethink mechanisms to address some issues of privacy and accuracy when data are either shared or exchanged before mining. Such mechanisms can lead to new privacy control methods to convert a database into a new one that conceals private information while preserving the general patterns and trends from the original database.

REFERENCES

Agrawal, R., & Srikant, R. (2000). Privacy-preserving data mining. In *Proceedings of the 2000 ACM SIGMOD international conference on management of data* (pp. 439-450). Dallas, Texas.

Brankovic, L & Estivill-Castro, V. (1999). Privacy Issues in Knowledge Discovery and Data Mining. In *Proceedings of Australian Institute of Computer Ethics Conference (AICEC99)*, Melbourne, Victoria, Australia, July, 1999.

Clifton, C., Kantarcioğlu, M., & Vaidya, J. (2002). Defining privacy for data mining. In *Proceedings of the National Science Foundation Workshop on Next Generation Data Mining* (pp. 126-133). Baltimore, MD, USA.

Clifton, C., & Marks, D. (1996). Security and privacy implications of data mining. *In Proceedings of the Workshop on Data Mining and Knowledge Discovery* (pp. 15-19). Montreal, Canada.

Cockcroft, S., & Clutterbuck, P. (2001). Attitudes towards information privacy. In *Proceedings of the 12th Australasian Conference on Information Systems. Coffs Harbour*. NSW, Australia.

Culnan, M. J. (1993). How did they get my name?: An exploratory investigation of consumer attitudes toward secondary information. *MIS Quartely, 17*(3), 341-363.

Estivill-Castro, V., & Brankovic, L. (1999). Data swapping: balancing privacy against precision in mining for logic rules. In *Proceedings of Data Warehousing and Knowledge Discovery DaWaK-99* (pp. 389-398). Florence, Italy.

Evfimievski, A., Srikant, R., Agrawal, R., & Gehrke, J. (2002). Privacy Preserving mining of association rules. In *Proceedings of the 8th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining* (pp. 217-228). Edmonton, AB, Canada.

Felty, A. P., & Matwin, S. (2002). Privacy-oriented data mining by proof checking. In *Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD)* (pp. 138-149). Helsinki, Finland.

Goldreich, O., Micali, S., & Wigderson, A. (1987). How to play any mental game - a completeness theorem for protocols with honest majority. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing* (pp. 218-229). New York City, USA.

Kantarcioğlu, M., & Clifton, C. (2002). Privacy-preserving distributed mining of association rules on horizontally partitioned data. In *Proceedings of The ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery*. Madison, Wisconsin.

Klösgen, W. (1995). KDD: Public and private concerns. *IEEE EXPERT, 10*(2), 55-57.

Lindell, Y., & Pinkas, B. (2000). Privacy preserving data mining. In *Crypto 2000, Springer-Verlag (LNCS 1880)* (pp. 36-54). Santa Barbara, CA.

Liu, K., Kargupta, H., & Ryan, J. (2006). Random projection-based multiplicative data perturbation for privacy preserving distributed data mining. *IEEE Transactions on Knowledge and Data Engineering, 18*(1), 92-106.

Meregu, S., & Ghosh, J. (2003). Privacy-preserving distributed clustering using generative models. In *Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM'03)* (pp. 211-218). Melbourne, Florida, USA.

Mucsi-Nagy, A., & Matwin, S. (2004). Digital fingerprinting for sharing of confidential data. In *Proceedings of the Workshop on Privacy and Security Issues in Data Mining* (pp. 11-26). Pisa, Italy.

O'Leary, D. E. (1995). Some privacy issues in knowledge discovery: The OECD personal privacy guidelines. *IEEE EXPERT, 10*(2), 48-52.

Oliveira, S. R. M., Zaïane, O. R., & Saygin, Y. (2004a). Secure association rule sharing. In *Proceedings of the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'04)* (pp. 74-85). Sydney, Australia.

Oliveira, S. R. M. Oliveira & Zaïane, O.R. (2004b, August) Toward Standardization in privacy-preserving data mining. In *Proceedings of the 3rd Workshop on Data Mining Standards (DM-SSP 2004), in conjunction with KDD 2004*. Seattle, WA, USA, (pp. 7-17).

Oliveira, S. R. M. & Zaïane, O. R. (2006). A unified framework for protecting sensitive association rules in business collaboration. *Int. J. Business Intelligence and Data Mining, 1*(3): 247-287.

Oliveira, S. R. M. & Zaïane, O. R. (2007a) Privacy-preserving clustering to uphold business collaboration: A dimensionality reduction-based transformation approach. *International Journal of Information Security and Privacy, 1*(2): 13-36.

Oliveira, S. R. M. & Zaïane, O. R. (2007b) A privacy-preserving clustering approach toward secure and effective data analysis for business collaboration. *Computers & Security, 26*(1): 81-93.

Piatetsky-Shapiro, G. (1995). Knowledge discovery in personal data vs. privacy: A mini-symposium. *IEEE Expert, 10*(2), 46-47.

Rizvi, S. J., & Haritsa, J. R. (2002). Maintaining data privacy in association rule mining. In *Proceedings of the 28th International Conference on Very Large Data Bases*. Hong Kong, China.

Saygin, Y., Verykios, V. S., & Clifton, C. (2001). Using unknowns to prevent discovery of association rules. *SIGMOD Record, 30*(4), 45-54.

Vaidya, J., & Clifton, C. (2002). Privacy preserving association rule mining in vertically partitioned data. In *Proceedings of the 8th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining* (pp. 639-644). Edmonton, AB, Canada.

Vaidya, J., & Clifton, C. (2003). Privacy-preserving k-means clustering over vertically partitioned data. In *Proceedings of the 9th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining* (pp. 206-215). Washington, DC, USA.

Veloso, A. A., Meira Jr., W., Parthasarathy, S., & Carvalho, M. B. (2003). Efficient, accurate and privacy-preserving data mining for frequent itemsets in distributed databases. In *Proceedings of the 18th Brazilian Symposium on Databases* (pp. 281-292). Manaus, Brazil.

Verykios, V. S., Elmagarmid, A. K., Bertino, E., Saygin, Y., & Dasseni, E. (2004). Association rule hiding. *IEEE Transactions on Knowledge and Data Engineering, 16*(4), 434-447.

Zang, N., Wang, S., & Zhao, W. (2004). A New Scheme on Privacy Preserving Association Rule Mining. In *Proceedings of the 15th European Conference on Machine Learning and the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases*. Pisa, Italy.

KEY TERMS

Collective Privacy: Is concerned with the protection of *sensitive knowledge* representing the activities of a group.

Data Sanitization: Is the process of hiding *sensitive rules* in transactional databases. The sanitization is achieved by modifying some transactions. In some cases, a number of items are deleted from a group of transactions with the purpose of hiding the sensitive rules derived from those transactions. In doing so, the support of such sensitive rules is decreased below a certain disclosure threshold defined by the data owner.

Disclosure Threshold: The process of hiding some *sensitive rules* satisfies a disclosure threshold controlled by the database owner.

Individual Privacy: Is concerned with the protection of personally identifiable information. In general, information is considered personally identifiable if it can be linked, directly or indirectly, to an individual person.

Privacy-Preserving Data Mining: Encompasses the dual goal of meeting privacy requirements and providing valid data mining results.

Sensitive Knowledge: Is described as the knowledge that can provide competitive advantage in the business world.

Sensitive Rules: Are a special group of association rules which represent the *sensitive knowledge* mined from databases.

Process Mining to Analyze the Behaviour of Specific Users

Laura Măruşter
University of Groningen, The Netherlands

Niels R. Faber
University of Groningen, The Netherlands

INTRODUCTION

As the on-line services and Web-based information systems proliferate in many domains of activities, it has become increasingly important to model user behaviour and personalization, so that these systems will appropriately address user characteristics. In this sense, particular topics are addressed by research in human-computer interaction (HCI), such as the discovering of user behaviour or navigation styles (Balajinath & Raghavan, 2001; Herder & Juvina, 2005; Juvina & Herder, 2005b; Mensalvas et al., 2003; Obendorf, et al., 2007), developing metrics involved in modelling and assessing web navigation (Herder, 2002; McEneany, 2001; Spiliopoulou & Pohle, 2001), and cognitive models for improving the redesign of information systems (Bollini, 2003; Ernst & Story, 2005; Juvina & Herder, 2005a; Juvina et al., 2005b; Lee & Lee, 2003).

Various methods have been developed to model web navigation in case of generic users (Eirinaki &Vazirgiannis, 2003). The existence of systems and/or interfaces neglecting specific user groups results into low performance of these systems, which requires further redesign. By investigating navigational patterns of specific user groups, and combining with their specific characteristics, the (re)design of the systems can be made more effectively. In this chapter, farmers have been considered as a specific user group. However, the methodology discussed in this chapter can be used also in case of other specific user groups.

Focusing on farmers as a specific IT user group, becomes an important research issue (Thysen, 2000). Farmers show a low rate of management software adoption (Alvarez & Nuthall, 2006). Different projects have been initiated to support farmers, to pursue their decision-making activities with the aid of Information Systems (see Fountas, Wulfsohn, Blackmore, Jacobsen, & Pederson, 2006; US North Central Research in Farm Information Systems, 2000). Kuhlmann & Brodersen (2001) and Hayman (2003) express their pessimism about the fast diffusion of complex information technology tools and decision support systems (DSS) among farmers. Various studies aimed to find factors that hamper adoption of DSSs in agriculture (Faber, Jorna, Van Haren, & Maruster, 2007; Kerr, 2004; Kuhlmann & Brodersen, 2001). Alvarez & Nuthall (2006) conclude that "software developers must work with farmers, both in design, and training and support, and the system must be configurable to suit a range of farmer characteristics". Therefore, there seems to be a real need to personalize these systems, such that they address farmers' characteristics.

Personalization of website design that supports DSS systems, to incorporate user characteristics, enhances effectiveness and usage of these systems. "The goal of personalization is to provide users with what they want or need without requiring them to ask for it explicitly" (Mulvenna, Anand, & Buchner, 2000).

The enhancement of website effectiveness is especially relevant in case a website offers access to underlying systems that aim to provide support and advice to its users. For instance, Jensen (2001) analyzed the usage of a web-based information system for variety selection in field crops. He compared four user groups by constructing measures based on logged information. This analysis reveals interesting similarities and differences in behaviour concerning the four groups. However, no insights could be given about the most typical sequence of behaviour/navigation patterns, such that it could support the redesign of the system.

In this chapter, a methodology of investigating user behaviour/navigation patterns by employing process mining technique is presented (Maruster & Faber, 2007). This methodology consists of (i) obtaining user groups considering some performance criteria, (ii) determining navigational patterns for all users and

for each user group, and (iii) deriving implications for redesign. In the following section, this methodology is presented, using an example involving starch potato farmers. We conclude this chapter with future trends and conclusions.

PROCESS MINING FOR PERSONALIZATION

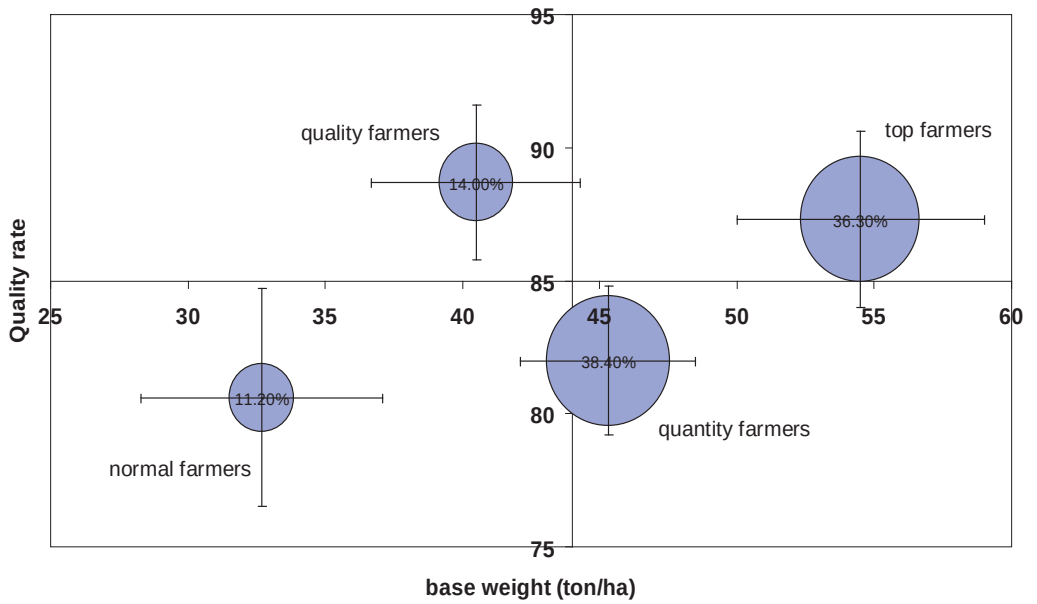
Web Usage Mining by exploiting web access logs targets General Access Pattern Tracking and Customized Usage Tracking. *General Access Pattern Tracking* is using KDD (Knowledge Discovery in Data) techniques to understand general access patterns and trends of web usage, while *Customized Usage Tracking* analyses access patterns of each user at a time (see Zaïane, 2007). Customized Usage Tracking can be used to provide recommendations to web users via personalization. Personalization can be done by distinguishing between different users or group of users, which is called *User profiling* (Eirinaki &Vazirgiannis, 2003).

Developing User Groups

Farmers are considered as a specific group of professional users, characterized by a modest use of IT. In the context of using applications (e.g. decision support systems) developed on Web-service platforms, personalization decisions are not straightforward and have to be based on knowledge about the user. Applications developed for farmers such as decision support systems (DSS) cannot consider them either as generic users (they have low experience with IT, lack of interest to use IT), or individual users (because DSS are complex systems that are hardly customizable for individual users).

Clustering is a common technique used in user profiling (Eirinaki &Vazirgiannis, 2003). Using two-step clustering method[1], Dutch starch potato farmers have been split up into four groups based on performance characteristics, where quantity and quality of a farmer's yield are the used dimensions for cluster analysis, both averaged over the last three years (Faber, Peters & Jorna, 2006). The found clusters have been labelled

Figure 1. Farmer clusters



respectively normal farmers, quality farmers, quantity farmers, and top farmers. In Figure 1, the size of the circles refers to the contribution of the respective clusters to the total production of starch (also indicated with the mentioned percentages).

Determining Navigational Patterns with Process Mining

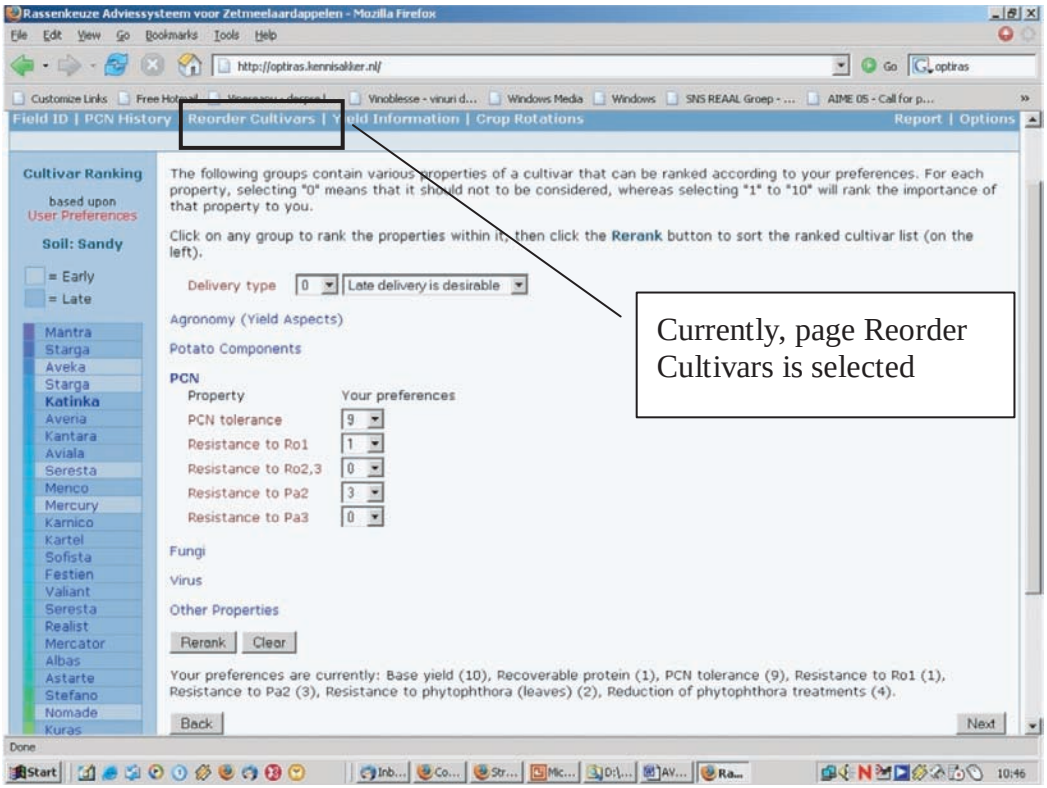
Different methods have been used to investigate the navigation patterns of users, by using logged data. For instance, Herder (2002) developed a technique based on aggregated measures, McEneany (2001) employed graphical and numerical methods, and Spiliopoulou & Pohle (2001) developed a mining query language for depicting navigational patterns.

Data mining techniques are used to develop navigation patterns (or to model user behaviour), such as Clustering, Classification, Association Rule Discovery, Sequential pattern Discovery, Markov models, hidden (latent) variable models, and hybrid models (Mobasher, 2006).

In this chapter, navigational patterns exhibited by farmers are determined using a technique called process mining. *Process mining* techniques allow for extracting information from event logs: for instance, transaction logs of an enterprise resource planning system can be used to discover models describing processes, organisations, and products (Van der Aalst et al., 2007). While navigational patterns developed with the above-mentioned data mining techniques are sequential, process mining may discover also parallel patterns. Moreover, it is possible to use process mining to monitor deviations (e.g. comparing the observed events with predefined models) (Rozinat & Van der Aalst, 2006)[2].

Maruster & Faber (2007) used process mining employing the logs resulted from using a decision support system called OPTIRas™, a DSS that aids starch potato farmers in their cultivar selection activities (Agrobio-

Figure 2. OPTIRas™ screenshot



kon, 2007). The interface of OPTIRas™ DSS consists on seven main pages: Field ID, PCN History, Reorder Cultivars, Yield Information, Crop Rotation, Report and Option (see Figure 2).

Farmers can login in OPTIRas registering with their e-mails, but they can register also with an anonymous ID. This manner of anonymousness provides advantages for the users, but brings disadvantages to the analysis. A user may login first time with his/her e-mail, the second time anonymously using computer A, and the third time again anonymously using computer B; in the analysis, these three sessions will count as sessions belonging to three separate users, and we cannot see anymore how a user eventually changes his behaviour in time.

Table 1 shows an excerpt from the log, which contains information concerning the movement from one 'source' page to a 'destination' page, such as time stamp, name of source page, name of destination page. The final log file consists on 763 user sessions, belonging to 501 individual users.

Navigational patterns are obtained with the aid of an algorithm called *genetic process mining* available as a plug-in the process mining ProM framework (ProM, 2007). Genetic based algorithms for process mining provide good heuristics that can tackle relevant structural constructs (i.e. sequences, parallelism, choices, loops). Moreover, these algorithms are equipped with analysis metrics that assess how complete, precise and folded the resulting process representations are (Medeiros, 2006).

First, the navigational patterns of all users are determined. In Figure 3 is shown the result of genetic process mining on the log including the sessions of all farmers. The focus is to grasp the patterns of navigation from one page to another ('FromPage' field was considered for mining, see Table 1). The rectangles in Figure 3 refer to transitions or activities, in our case to page names, such as Field, Order, etc. There are two special activity names, ArtificialStartTask and ArtificialEndTask that refer to a generic start or end page. The term 'complete' refers to a finished action. The number inside the rectangle shows how many times a page has been invoked. The arcs (associated with a number) between rectangles represent the occurrence of two pages and how often it happens. For instance in Figure 3, Pcn page occurred 526 times after Field page.

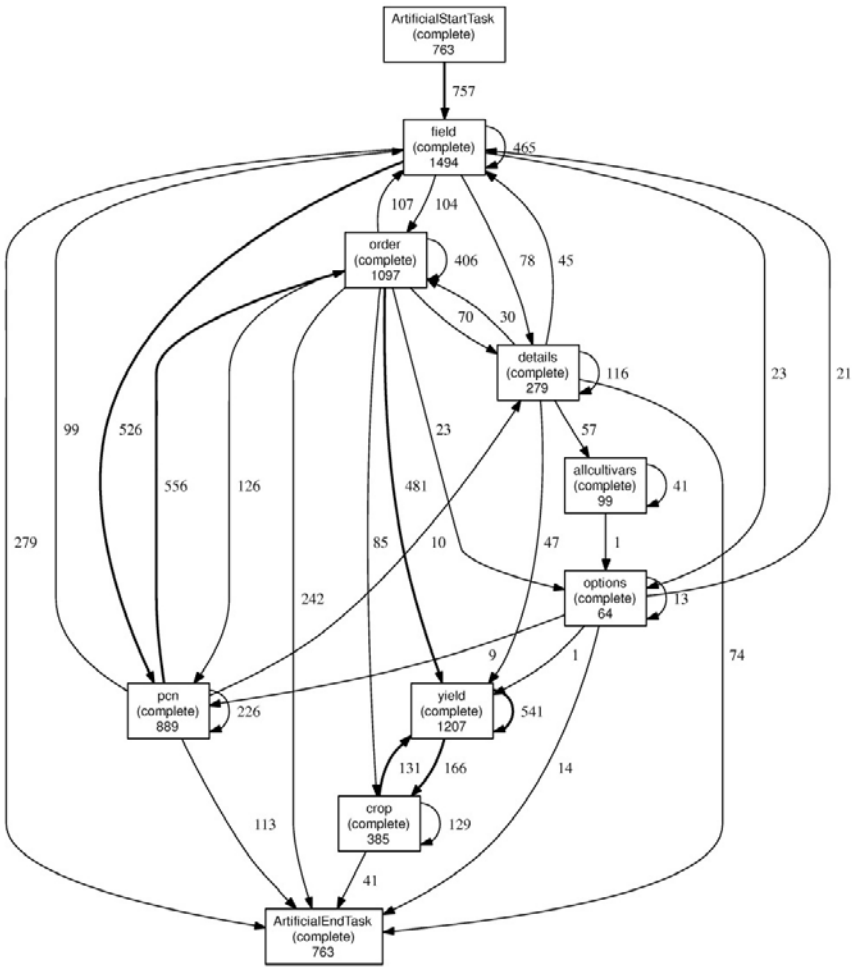
In Figure 3 we can determine "the most common navigational pattern" (represented with bold arrows) which consists of ArtificialStartTask→Field (757), Field→Pcn (526), Pcn→Order (556), Order→Yield (481), Yield→ Crop (166), Yield→ Yield (541), Crop→ Yield (131). The sequence Field → Pcn → Order → Yield → Crop is actually the "prescribed" order of pages in OPTIRas, e.g. the order in which pages are presented in OPTIRas.

Second, the ProM genetic process mining is used on the logs including 22 sessions of top-farmers, 8 belonging to quality farmers, and 8 to quantity farmers (there were not sessions belonging to normal farmers).

Table 1. Excerpt from a log file

Time stamp	**FromPage**	**ToPage**
2004-12-22 22:13:29	field	Pcn
2004-12-22 22:13:35	pcn	Order
2004-12-22 22:14:00	order	Yield
2004-12-22 22:14:26	yield	Crop
2004-12-22 22:16:16	crop	Yield
2004-12-22 22:16:25	yield	Crop
2004-12-22 22:16:53	crop	details (katinka_l)
2004-12-22 22:17:54	details	Allcultivars

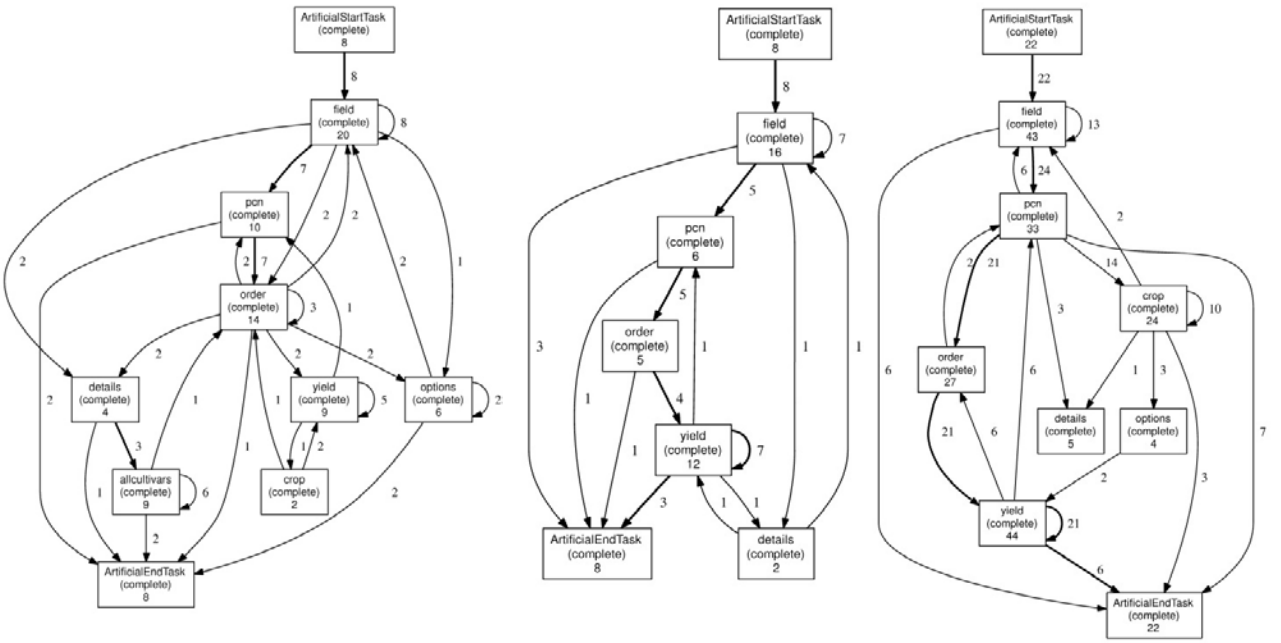
Figure 3. The mined behaviour of all farmers (run of the genetic algorithm)



This analysis shows a distinction of three types of users (see Figure 4). Quality farmers show "spread" behaviour, without clear defined patterns (Figure 4a). Top farmers are seeking information and looping (Field→ Pcn→Field, Pcn→Order→Pcn, Order→Yield→Order, Pcn→Crop→Field) (Figure 4b). Quantity farmers seem to follow in majority the prescribed order of pages (emphasized with bold arrows), e.g. ArtificialStartTask → Field → Pcn → Order → Yield→ ArtificialEndTask (Figure 4c).

This methodology shows how to combine navigational patterns with already identified user groups, which allows for the construction of navigational patterns for specific groups of users. Top-farmers show the most complex behavioural pattern of usage, visiting webpages in a non-linear order and backtracking their steps. In contrast, quality farmers have navigational patterns that simply comply with the prescribed sequence of pages of the website.

Figure 4. The mined behaviour for 'quality farmers' (a), 'top farmers' (b) and 'quantity farmers' (c), using genetic process mining



(a) *(b)* *(c)*

Implications for Redesign

The combination of navigational patterns and user characteristics can be used to redesign websites. In Maruster & Faber (2007), based on the identified behaviours of top, quantity, and quality farmers, three distinct behaviours regarding the DSS have been identified. Because these are preliminary findings, a detailed redesign cannot be made yet. However, from current findings two possibilities for redesign are inferred. First, because quantity farmers show navigational behaviour aligned with the current design of the website, no redesign seems to be needed for this particular group. Second, because top farmers shows more complex behaviour, a possibility for redesign is to group more information and functions on one page (Faber, Peters, & Jorna, 2006). In this manner, users in this group will be able to see the consequences of their actions directly, without having to go back and forth through various webpages.

FUTURE TRENDS

Although research in personalization results in many successful contributions, various challenges still lie ahead. Modelling user context and interests, is one of challenges existing in the domain of personalization (Mobasher, 2006). Determining the cognitive models and learning styles of users will provide the necessary information to understand better the users, their decision-making process, interests and needs, which subsequently will lead to improvements of user models. Pieters (2005) found that farmer clusters differ significantly on the motivation scales extrinsic value and self-efficacy and on the skills scales peer learning

and help seeking. Top farmers appear to have better learning attitudes relative to other farmer clusters. Quality oriented farmers seem to stay behind with respect to their learning skills, especially on the scales of peer learning, and help seeking. These farmers might perform better when their learning skills are addressed properly. The normal farmers rank very high regarding extraversion and openness. This opens the possibility that these farmers are not highly interested in improving starch potato cultivation but have their interests in other aspects of their farm or life. Learning styles and personality have impact on use of DSS's; especially anxiety is one of the scales, which blocks effectively the usage of DSS's. All these factors may be taken into account when redesigning and/or personalizing a DSS.

CONCLUSION

Mining user behaviour for personalization enables the redesign for personalization of existing websites, resulting in websites that are aligned with actual usage behaviour. The split up of users based on their characteristics results in an initial identification of different user groups. In the domain of crop selection, application users are farmers. History has shown that farmers cannot be treated as one homogeneous group, and instead should be treated individually. However, decision support systems for these farmers are complex systems, making the tailoring of these systems to individual farmers difficult. Therefore, the population of farmers has been split up into various groups based on performance characteristics, resulting in four different groups.

Process mining techniques are used to construct navigational patterns from user logs. The level of detail of user behaviour that can be mined depends on the granularity of the logs. From the user logs individual and aggregated behavioural patterns can be constructed. Combining navigational patterns with already identified user groups allows for the construction of navigational patterns for specific groups of users. Ultimately, the combination enables the redesign of the existing application, tailored for each user group. The application can be tailored to fit specific characteristics of the different user groups and their particular behaviour.

REFERENCES

Agrobiokon (2007). OPTIRas: Rassenkeuze Adviessysteem voor Zetmeelaardappelen. Retrieved 15-3-2007, from http://optiras.agrobiokon.eu/

Alvarez, J. & Nuthall, P. (2006). Adoption of computer based information systems. The case of dairy farmers in Canterbury, NZ, and Florida, Uruguay, Computers and Electronics in Agriculture, 50, 48-60.

Balajinath, B. & Raghavan, S. (2001). Intrusion detection through learning behavior model. Computer Communication, 24, 1202-1212.

Bollini, L. (2003). Web Interface Design based on Cognitive Maps: Generative Dynamics in Information Architecture. Retrieved 15-3-2007, from http://www.generativeart.com/papersga2003/a07.htm

Eirinaki, M. & Vazirgiannis, M. (2003). Web Mining for Web Personalization, ACM Transactions on Internet Technologies, 3, 1-27.

Ernst, N. A. & Story, M.-A. (2005). Cognitive support for ontology modeling. International Journal of Human-computer Studies, 62, 553-577.

Faber, N.R., Jorna, R.J., van Haren, R.J.F., & Maruster, L. (2007). Knowledge and knowledge use for sustainable innovation: the case of starch potato production; achieving more with less. Proceedings of the 10th International Conference on Organisational Semiotics 2007, 24-26 July, Sheffield, p.54-65.

Faber, N.R., Peters, K. & Jorna, R.J. (2006). Technology for knowledge crossover: a tool for a sustainable paper industry. Proceedings of the 9th International Conference on Organisational Semiotics.

Fountas, S., Wulfsohn, D., Blackmore, B. S., Jacobsen, H. L., & Pederson, S. M. (2006). A model of decision-making and informations flows for infomation-intensive agriculture. Agricultural Systems, 87, 192-210.

Hayman, P. (2003). Decision support systems in Australian dryland farming: A promising past, a disspointing present and uncertain future. Retrieved 15-3-2007, from http://www.cropscience.org.au/icsc2004/pdf/1778_haymanp.pdf

Herder, E. (2002). Metrics for the adaptation of site structure. In Proceedings of the German Workshop on

Adaptivity and User Modeling in Interactive Systems ABIS '02 (pp. 22-26). Hannover.

Herder, E. & Juvina, I. (2005). Discovery of individual user navigation styles. In Proceedings of the Workshop on Individual Differences - Adaptive Hypermedia. Eindhoven.

Jensen, A. L. (2001). Building a web-based information system for variety selection in field crops-objectives and results. Computers and Electronics in Agriculture, 32, 195-211.

Juvina, I. & Herder, E. (2005a). Bringing Cognitive Models into the Domain of Web Accesibility. In Proceedings of HCII 2005 Conference.

Juvina, I. & Herder, E. (2005b). The Impact of Link Suggestion on User Navigation and User Perception. In Proceedings of User Modelling 2005 (pp. 483-492). Berlin: Springer-Verlag.

Kerr, D. (2004). Factors influencing the Development and Adoption of Knowledge Based Decision Support Systems for Small, Owner-Operated Rural Business. Artificial Intelligence Review, 22, 127-147.

Kuhlmann, F. & Brodersen, C. (2001). Information Technology and farm management: development and perspectives. Computers and Electronics in Agriculture, 30, 71-83.

Lee, K. C. & Lee, S. (2003). A cognitive map simulation approach to adjusting the design factors of the electronic commerce web sites. Expert Systems with Applications, 24, 1-11.

Maruster, L. & Faber, N. R. (2007). A process mining approach to analysing user behaviour. Retrieved 15-3-2007, from http://www.duurzameinnovatie.nl/Files/procmining_2007.pdf

McEneany, J. E. (2001). Graphic and numerical methods to assess navigation in hypertext. International Journal of Human-computer Studies, 55, 761-786.

Medeiros, A.K. (2006). Genetic Process Mining, PhD thesis, University of Einhoven, The Netherlands.

Mensalvas, E., Millan, S., Perez, M., Hochsztain, E., Robles, V., Marban, O. et al. (2003). Beyond user clicks: an algoritm and an agent-based arhitecture to discover user behavior. In Proceedings of ECML/PKDD, First European Web Mining Forum. Berlin: Springer-Verlag.

Mobasher, B. (2006). Data Mining for Personalization. In P.Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), The Adaptive Web: Methods and Strategies of Web Personalization. Berlin: Springer-Verlag.

Mulvenna, M., Anand, S. S., & Buchner, A. G. (2000). Personalization on the net using web mining. Communications of the ACM, 43, 122-125.

Obendorf, H., Weinreich, H., Herder, E. & Mayer, M. (2007). Web Page Revisitation Revisited: Implications of a Long-term Click-stream Study of Browser Usage. In Proceedings of the SIGCHI conference on Human factors in computing systems.

Pieters, D. (2005). Stijlvol leren en kennis overdragen: een onderzoek naar leer- en persoonlijkheidsstijlen bij zetmeelaardappeltelers van AVEBE, Master thesis, University of Groningen.

ProM (2007). Process Mining website. Retrieved 26.07.2007, from http://www.processmining.org.

Rozinat, A. & Van der Aalst, W,M.P. (2006). Conformance Testing: Measuring the Fit and Appropriateness of Event Logs and Process Models, LNCS (3812), 163–176.

Spiliopoulou, M. & Pohle, C. (2001). Data Mining to Measure and Improve the Success of web Sites. Data Mining and Knowledge Discovery, 5, 85-114.

Thysen, I. (2000). Agriculture in the information society. Journal of Agricultural Engineering, 76, 297-303.

US North Central Research in Farm Information Systems (2000). Farm Information Systems: Their development and use in decision-making (Rep. No. 345). Ames, IS: Iowa State University.

Van der Aalst, W.M.P., Reijers, H.A, Weijters, A.J.M.M., van Dongen, B.F., Alves de Medeiros, A.K., Song, M., & Verbeek, H.M.W. (2007), Business process mining: An industrial application, Information systems, 32, 713-732.

Zaïane, O. R. (2007). Web mining taxonomy. Retrieved 15-3-2007, from http://www.cs.ualberta.ca/~zaiane/courses/cmput690/slides/Chapter9/sld012.htm

KEY TERMS

Decision Support System: An application, possibly accessible through a website that informs its user regarding a specific domain, enabling the user to effectively make decisions in relation to his tasks.

Human-Computer Interaction: Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them.

Navigational Pattern: Represent the manner in which a user or a group of users are navigating through a web site.

Process Mining: Allow for extracting information from event logs. These techniques can be used to discover models describing processes, organisations, and products. Moreover, it is possible to use process mining to monitor deviations (e.g., comparing the observed events with predefined models or business rules).

Specific User Group: A cluster of people that is identified based on one or more characteristics that the people of the cluster have in common (for instance, growers can be split into various clusters based on yield performance measures).

System Redesign: Altering the design of an existing software application or part thereof, by revising a function or the appearance of the application or by making a new design for the application.

User Behaviour: The observable result of the reasoning of a human-being who uses a computer application that is logged.

User Profiling: Distinguishing between users or group of users, by creating an information base that contains the preferences, characteristics, and activities of the users.

ENDNOTES

1 Procedure available in SPSS software, Release 12.

2 For more details about process mining, visit http:/www.processmining.org

Profit Mining

Senqiang Zhou
Simon Fraser University, Canada

Ke Wang
Simon Fraser University, Canada

INTRODUCTION

A major obstacle in data mining applications is the gap between the statistic-based pattern extraction and the value-based decision-making. "Profit mining" aims to reduce this gap. In profit mining, given a set of past transactions and pre-determined target items, we like to build a model for recommending target items and promotion strategies to new customers, with the goal of maximizing profit. Though this problem is studied in the context of retailing environment, the concept and techniques are applicable to other applications under a general notion of "utility". In this short article, we review existing techniques and briefly describe the profit mining approach recently proposed by the authors. The reader is referred to (Wang, Zhou & Han, 2002) for the details.

BACKGROUND

It is a very complicated issue whether a customer buys a recommended item. Consideration includes items stocked, prices or promotions, competitors' offers, recommendation by friends or customers, psychological issues, conveniences, etc. For on-line retailing, it also depends on security consideration. It is unrealistic to model all such factors in a single system. In this article, we focus on one type of information available in most retailing applications, namely past transactions. The belief is that shopping behaviors in the past may shed some light on what customers like. We try to use patterns of such behaviors to recommend items and prices.

Consider an on-line store that is promoting a set of *target items*. At the cashier counter, the store likes to recommend one target and a promotion strategy (such as a price) to the customer based on *non-target items* purchased. The challenge is determining an item interesting to the customer at a price affordable to the customer and profitable to the store. We call this problem *profit mining* (Wang, Zhou & Han, 2002).

Most statistics-based rule mining, such as association rules (Agrawal, Imilienski & Swami, 1993; Agrawal & Srikant, 1994), considers a rule as "interesting" if it passes certain statistical tests such as support/confidence. To an enterprise, however, it remains unclear how such rules can be used to maximize a given business object. For example, knowing "Perfume→ Lipstick" and "Perfume→Diamond", a store manager still cannot tell which of Lipstick and Diamond, and what price should be recommended to a customer who buys Perfume. Simply recommending the most profitable item, say Diamond, or the most likely item, say Lipstick, does not maximize the profit because there is often an inverse correlation between the likelihood to buy and the dollar amount to spend. This inverse correlation reflects the general trend that the more dollar amount is involved, the more cautious the buyer is when making a purchase decision.

MAIN THRUST OF THE CHAPTER

Related Work

Profit maximization is different from the "hit" maximization as in classic classification because each hit may generate different profit. Several approaches existed to make classification *cost-sensitive*. (Domingos, 1999) proposed a general method that can serve as a wrapper to make a traditional classifies cost-sensitive. (Zadrozny & Elkan, 2001) extended the error metric by allowing the cost to be example dependent. (Margineantu & Dietterich, 2000) gave two bootstrap methods to estimate the average cost of a classifier. (Pednault, Abe & Zadrozny, 2002) introduced a method to make sequential cost-sensitive decisions, and the goal is to maximize the total

benefit over a period of time. These approaches assume a given error metric for each type of misclassification, which is not available in profit mining.

Profit mining is related in motivation to *actionability* (or *utility*) of patterns: a pattern is interesting in the sense that the user can act upon it to her advantage (Silberschatz & Tuzhilin, 1996). (Kleinberg, Papadimitriou & Raghavan, 1998) gave a framework for evaluating data mining operations in terms of utility in decision-making. These works, however, did not propose concrete solutions to the actionability problem. Recently, there were several works applying association rules to address business related problems. (Brijs, Swinnen, Avanhoof & Wets, 1999; Wong, Fu & Wang, 2003; Wang & Su, 2002) studied the problem of selecting a given number of items for stocking. The goal is to maximize the profit generated by selected items or customers. These works present one important step beyond association rue mining, i.e., addressing the issue of converting a set of individual rules into a single actionable model for recommending actions in a given scenario.

There were several attempts to generalize association rules to capture more semantics, e.g., (Lin, Yao & Louie, 2002; Yao, Hamilton & Butz, 2004; Chan, Yang & Shen, 2003). Instead of a uniform weight associated with each occurrence of an item, these works associate a general weight with an item and mine all itemsets that pass some threshold on the aggregated weight of items in an itemset. Like association rule mining, these works did not address the issue of converting a set of rules or itemsets into a model for recommending actions.

Collaborative filtering (Resnick & Varian, 1997) makes recommendation by aggregating the "opinions" (such as rating about movies) of several "advisors" who share the taste with the customer. Built on this technology, many large commerce web sites help their customers to find products. For example, Amazon.com uses "Book Matcher" to recommend books to customers; Moviefinder.com recommends movies to customers using "We Predict" recommender system. For more examples, please refer to (Schafer, Konstan & Riedl, 1999). The goal is to maximize the hit rate of recommendation. For items of varied profit, maximizing profit is quite different from maximizing hit rate. Also, collaborative filtering relies on carefully selected "item endorsements" for similarity computation, and a good set of "advisors" to offer opinions. Such data are not easy to obtain. The ability of recommending prices, in addition to items, is another major difference between profit mining and other recommender systems.

Another application where data mining is heavily used for business targets is *direct marketing*. See (Ling & Li, 1998**;** Masand & Shapiro, 1996; Wang, Zhou, Yeung & Yang, 2002), for example. The problem is to identify buyers using data collected from previous campaigns, where the product to be promoted is usually fixed and the best guess is about who are likely to buy. The profit mining, on the other hand, is to guess the best item and price for a given customer. Interestingly, these two problems are closely related to each other. We can model the direct marketing problem as profit mining problem by including customer demographic data as part of her transactions and including a special target item NULL representing no recommendation. Now, each recommendation of a non-NULL item (and price) corresponds to identifying a buyer of the item. This modeling is more general than the traditional direct marketing in that it can identify buyers for more than one type of item and promotion strategies.

Profit Mining

We solve the profit mining by extracting patterns from a set of past transactions. A transaction consists of a collection of sales of the form (item, price). A simple price can be substituted by a "promotion strategy", such as "buy one get one free" or "X quantity for Y dollars", that provides sufficient information for derive the price. The transactions were collected over some period of times and there could be several prices even for the same item if sales occurred at different times. Given a collection of transactions, we find *recommendation rules* of the form $\{s_1, \ldots, s_k\} \rightarrow \langle I,P \rangle$, where I is a target item and P is a price of I, and each s_i is a pair of non-target item and price. An example is (*Perfume, price=$20*)→(*Lipstick, price=$10*). This recommendation rule can be used to recommend Lipstick at the price of $10 to a customer who bought Perfume at the price of $20. If the recommendation leads to a sale of Lipstick of quantity Q, it generates $(10\text{-}C)*Q$ profit, where C is the cost of Lipstick.

Several practical considerations would make recommendation rules more useful. First, items on the left-hand side in s_i can be item categories instead to capture category-related patterns. Second, a customer may have paid a higher price if a lower price was not available at the shopping time. We can incorporate the

domain knowledge that paying a higher price implies the willingness of paying a lower price (for exactly the same item) to search for stronger rules at lower prices. This can be done through multi-level association mining (Srikant and Agrawal, 1995; Han and Fu, 1995), by modeling a lower price as a more general category than a higher price. For example, the sale {<*chicken, $3.8*>} in a transaction would match any of the following more general sales in a rule: <*chicken, $3.8*>, <*chicken, $3.5*>, <*chicken, $3.0*>, *chicken, meat, food.* Note that the last three sales are generalized by climbing up the category hierarchy and dropping the price.

A key issue is how to make a set of individual rules work as a single recommender. Our approach is ranking rules the *recommendation profit*. The recommendation profit of a rule *r* is defined as the average profit of the target item in r among all transactions that match *r*. Note that the rank by average profit implicitly takes into account of both confidence and profit because a high average profit implies that both confidence and profit are high. Given a new customer, we pick up the highest ranked matching rule to make recommendation.

Before making recommendation, however, "over-fitting" rules that work only for observed transactions, but not for new customers, should be pruned because our goal is to maximize profit on new customers. The idea is as follows. Instead of ranking rules by observed profit, we rank rules by *projected profit,* which is based on the *estimated error* of a rule adapted for pruning classifiers (Quinlan, 1993). Intuitively, the estimated error will increase for a rule that matches a small number of transactions. Therefore, over-fitting rules tend to have a larger estimated error, which translates into a lower projected profit, and a lower rank.

For a detailed exposure and experiments on real life and synthetic data sets, the reader is referred to (Wang, Zhou & Han, 2002).

FUTURE TRENDS

The profit mining proposed is only the first, but important, step in addressing the ultimate goal of data mining. To make profit mining more practical, several issues need further study. First, it is quite likely that the recommended item tends to be the item that the customer will buy independently of the recommendation. Obviously, such items need not be recommended, and recommendation should focus on those items that the customer may buy if informed, but may not otherwise. Recommending such items likely brings in *additional* profit. Second, the current model maximizes only the profit of "one-shot selling effect," therefore, a sale in a large quantity is favored. In reality, a customer may regularly shop the same store over a period of time, in which case a sale in a large quantity will affect the shopping frequency of a customer, thus, profit. In this case, the goal is maximizing the profit for reoccurring customers over a period of time. Another interesting direction is to incorporate the feedback whether a certain recommendation is rejected and accepted to improve future recommendations.

This current work has focused on the information captured in past transactions. As pointed out in Introduction, other things such as competitors' offers, recommendation by friends or customers, consumer fashion, psychological issues, conveniences, etc. can affect the customer's decision. Addressing these issues requires additional knowledge, such as competitors' offers, and computers may not be the most suitable tool. One solution could be suggesting several best recommendations to the domain expert, the store manager or sales person in this case, who makes the final recommendation to the customer after factoring the other considerations.

CONCLUSION

Profit mining is a promising data mining approach because it addresses the ultimate goal of data mining. In this article, we study profit mining in the context of retailing business, but the principles and techniques illustrated should be applicable to other applications. For example, "items" can be general actions and "prices" can be a notion of utility resulted from actions. In addition, "items" can be used to model customer demographic information such as Gender, in which case the price component is unused.

REFERENCES

Agrawal, R., Imilienski, T., & Swami, A. (1993, May). Mining association rules between sets of items in large databases. *ACM Special Interest Group on Management of Data (SIGMOD)* (pp. 207-216), Washington D.C., USA.

Agrawal, R. & Srikant, R. (1994, September). Fast algorithms for mining association rules. *International Conference on Very Large Data Bases (VLDB)* (pp. 487-499), Santiago de Chile.

Brijs, T., Swinnen, G., Vanhoof, K., & Wets, G. (1999, August). Using association rules for product assortment decisions: A case study. *International Conference on Knowledge Discovery and Data Mining (KDD)* (pp. 254-260), San Diego, USA.

Chan, R., Yang, Q., & Shen, Y. (2003, November). Mining high utility itemsets. *IEEE International Conference on Data Mining (ICDM)* (pp. 19-26), Melbourne, USA.

Domingos, P. (1999, August). MetaCost: A general method for making classifiers cost-sensitive. *ACM SIG International Conference on Knowledge Discovery and Data Mining (SIGKDD)* (pp. 155-164), San Diego, USA.

Han, J., & Fu, Y. (1995, September). Discovery of multiple-level association rules from large databases. International Conference on Very Large Data Bases *(VLDB)* (pp. 420-431), Zurich, Switzerland.

Kleinberg, J., Papadimitriou, C. & Raghavan, P. (1998, December). A microeconomic view of data mining. *Data Mining and Knowledge Discovery Journal, 2*(4), 311-324.

Lin, T. Y., Yao, Y.Y., & Louie, E. (2002, May). Value added association rules. *Advances in Knowledge Discovery and Data Mining, 6th Pacific-Asia Conference PAKDD* (pp. 328-333), Taipei, Taiwan.

Ling, C., & Li, C. (1998, August) Data mining for direct marketing: problems and solutions. *ACM SIG International Conference on Knowledge Discovery and Data Mining (SIGKDD)* (pp. 73-79), New York, USA.

Margineantu, D. D., & Dietterich, G. T. (2000, June-July). Bootstrap methods for the cost-sensitive evaluation of classifiers. *International Conference on Machine Learning (ICML)* (pp. 583-590), San Francisco, USA.

Masand, B., & Shapiro, G. P. (1996, August) A comparison of approaches for maximizing business payoff of prediction models. *ACM SIG International Conference on Knowledge Discovery and Data Mining (SIGKDD)* (pp. 195-201), Portland, USA.

Pednault, E., Abe, N., & Zadrozny, B. (2002, July). Sequential cost-sensitive decision making with reinforcement learning. *ACM SIG International Conference on Knowledge Discovery and Data Mining (SIGKDD)* (p. 259-268), Edmonton, Canada.

Quinlan, J.R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann.

Resnick, P., & Varian, H.R. (1997). CACM special issue on recommender systems. *Communications of the ACM, 40*(3), 56-58.

Schafer, J. B., Konstan, J. A., & Riedl, J. (1999, November). Recommender systems in E-commerce. *ACM Conference on Electronic Commerce* (pp. 158-166), Denver, USA.

Silberschatz, A., & Tuzhilin, A. (1996). What makes patterns interesting in knowledge discovery systems. *IEEE Transactions on Knowledge and Data Engineering, 8*(6), 970-974.

Srikant, R., & Agrawal, R. (1995, September). Mining generalized association rules. *International Conference on Very Large Data Bases (VLDB)* (pp. 407-419), Zurich, Switzerland.

Wang, K., & Su, M. Y. (2002, July). Item selection by hub-authority profit ranking. *ACM SIG International Conference on Knowledge Discovery and Data Mining (SIGKDD)* (pp. 652-657), Edmonton, Canada.

Wang, K., Zhou, S., & Han, J. (2002, March). Profit mining: From patterns to actions. *International Conference on Extending Database Technology (EDBT)* (pp. 70-87), Prague, Czech Republic.

Wang, K., Zhou, S., Yeung, J. M. S., & Yang, Q. (2003, March). Mining customer value: From association rules to direct marketing. *International Conference on Data Engineering* (*ICDE)* (pp. 738-740), Bangalore, India.

Wong, R. C. W., Fu, A. W. C., & Wang, K. (2003, November). MPIS: Maximal-profit item selection with cross-selling considerations. *IEEE International Conference on Data Mining (ICDM)* (pp. 371-378), Melbourne, USA.

Yao, H., Hamilton, H. J., & Butz, C. J. (2004, April). A foundational approach for mining itemset utilities from databases. *SIAM International Conference on Data Mining (SIAMDM)* (pp. 482-486), Florida, USA.

Zadrozny, B., & Elkan, C. (2001, August). Learning and making decisions when costs and probabilities are both unknown. *ACM SIG International Conference on Knowledge Discovery and Data Mining (SIGKDD)* (pp. 204-213), San Francisco, USA.

KEY TERMS

Association Rule: An association has the form $I_1 \rightarrow I_2$, where I_1 and I_2 are two itemsets. The support of an association rule is the support of the itemset $I_1 \cup I_2$, and the *confidence* of a rule is the ratio of support of $I_1 \cup I_2$ and the support of I_1.

Classification: Given a set of training examples in which each example is labeled by a class, build a model, called a classifier, to predict the class label of new examples that follow the same class distribution as training examples. A classifier is accurate if the predicted class label is the same as the actual class label.

Cost Sensitive Classification: The error of a misclassification depends on the type of the misclassification. For example, the error of misclassifying Class 1 as Class 2 may not be the same as the error of misclassifying Class 1 as Class 3.

Frequent Itemset: The support of an itemset refers to as the percentage of transactions that contain all the items in the itemset. A frequent itemset is an itemset with support above a pre-specified threshold.

Over-fitting Rule: A rule has high performance (e.g. high classification accuracy) on observed transaction(s) but performs poorly on future transaction(s). Hence, such rules should be excluded from the decision-making systems (e.g. recommender). In many cases over-fitting rules are generated due to the noise in data set.

Profit Mining: In a general sense, profit mining refers to data mining aimed at maximizing a given objective function over decision making for a targeted population (Wang, Zhou & Han, 2002). Finding a set of rules that pass a given threshold on some interestingness measure (such as association rule mining or its variation) is not profit mining because of the lack of a specific objective function to be maximized. Classification is a special case of profit mining where the objective function is the accuracy and the targeted population consists of future cases. This paper examines a specific problem of profit mining, i.e., building a model for recommending target products and prices with the objective of maximizing net profit.

Transaction: A transaction is some set of items chosen from a fixed alphabet.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 930-934, copyright 2005 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

Program Comprehension through Data Mining

Ioannis N. Kouris
University of Patras, Dept. of Computer Engineering and Informatics Greece

INTRODUCTION

Software development has various stages, that can be conceptually grouped into two phases namely development and production (Figure 1). The development phase includes requirements engineering, architecting, design, implementation and testing. The production phase on the other hand includes the actual deployment of the end product and its maintenance. Software maintenance is the last and most difficult stage in the software lifecycle (Sommerville, 2001), as well as the most costly one. According to Zelkowitz, Shaw and Gannon (1979) the production phase accounts for 67% of the costs of the whole process, whereas according to Van Vliet (2000) the actual cost of software maintenance has been estimated at more than half of the total software development cost.

The development phase is critical in order to facilitate efficient and simple software maintenance. The earlier stages should be done by taking into consideration apart from any functional requirements also the later maintenance task. For example the design stage should plan the structure in a way that can be easily altered. Similarly, the implementation stage should create code that can be easily read, understood, and changed, and should also keep the code length to a minimum. According to Van Vliet (2000) the final source code length generated is the determinant factor for the total cost during maintenance, since obviously the less code is written the easier the maintenance becomes.

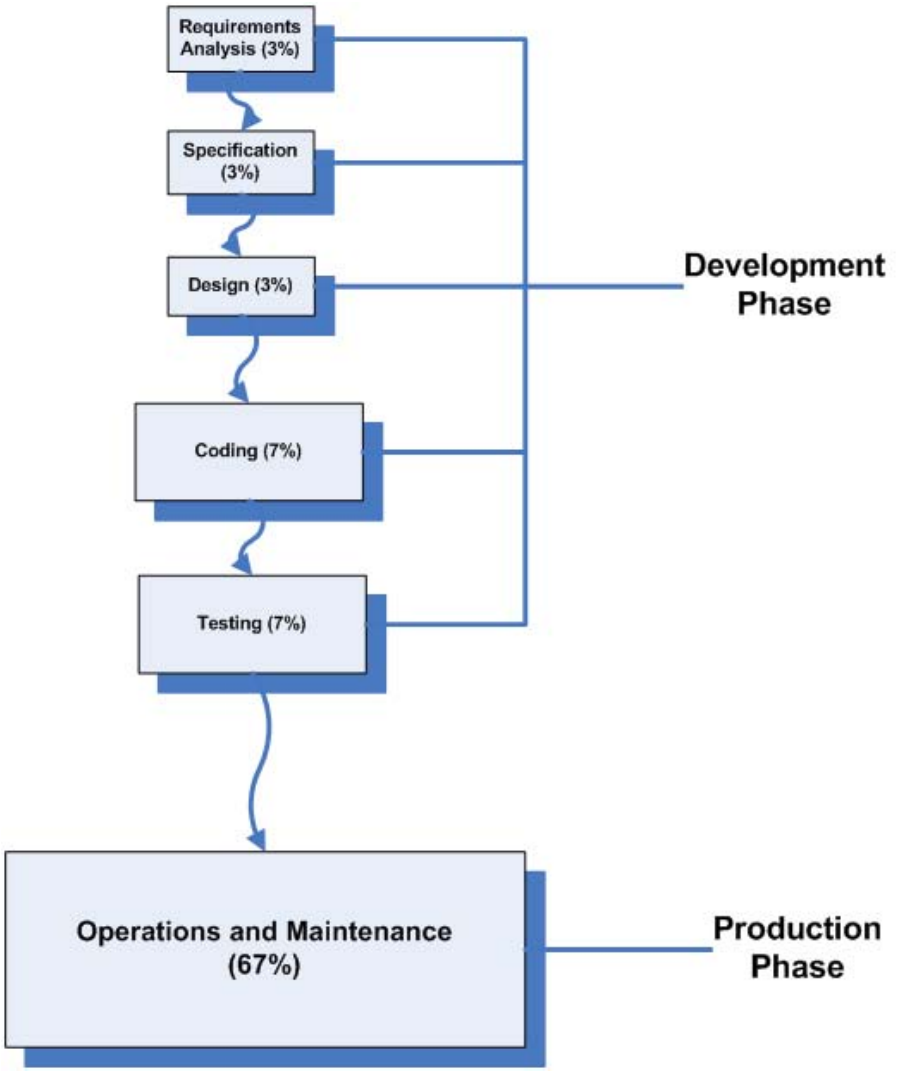
According to Erdil et al. (2003) there are four major problems that can slow down the whole maintenance process: unstructured code, maintenance programmers having insufficient knowledge of the system, documentation being absent, out of date, or at best insufficient, and software maintenance having a bad image. Thus the success of the maintenance phase relies on these problems being fixed earlier in the life cycle. In real life however when programmers decide to perform some maintenance task on a program such as to fix bugs, to make modifications, to create software updates etc. these are usually done in a state of time and commercial pressures and with the logic of cost reduction, thus finally resulting in a problematic system with ever increased complexity. As a consequence the maintainers spend from 50% up to almost 90% of their time trying to comprehend the program (Erdös and Sneed; 1998, Von Mayrhauser and Vans; 1994, Pigoski, 1996). Providing maintainers with tools and techniques to comprehend the programs has become and is receiving a lot of financial and research interest given the widespread of computers and software in all aspects of life. In this work we briefly present some of the most important techniques proposed in the field thus far and focus primarily on the use of data mining techniques in general and especially on association rules. Accordingly we give some possible solutions to problems faced by these methods.

BACKGROUND

Data mining can be defined as the process concerned with applying computational techniques (i.e. algorithms implemented as computer programs) to find patterns in the data. Among others, data mining technologies include association rule discovery, classification, clustering, summarization, regression and sequential pattern discovery (Chen, Han & Yu, 1996).

The use of data mining techniques in program comprehension has been very wide, with clustering being the most popular method. Tjortjis and Layzel (2001) have proposed a tool called DMCC (Data Mining Code Clustering) in order to help maintainers who are not familiar with some software to get a quick overview and speed up the process of getting familiar with it. The specific tool used clustering techniques for grouping together entities that share common characteristics. Mancoridis et al. (1998) have proposed a tool called Bunch that creates a high level decomposition of the structure of a system into meaningful subsystems by using clustering techniques over a Module Dependency Graph. Subsystems provide developers with high-level structural information that helps them navigate through

Figure 1. Software development stages and their relative costs (Zelkowitz, Shaw and Gannon, 1979)



the numerous software components, their interfaces, and their interconnections. Kanellopoulos and Tjortjis (2004) have also used clustering techniques on source code in order to facilitate program comprehension. Their work extracted data from C++ source code, and tried to cluster it in blocks in order to identify logical, behavioral and structural correlations amongst program components.

Other similar works can be found at Anquetil and Lethbridge (1999), Lakhotia (1997) and Tzerpos and Holt (1998). Also an early but thorough overview on software clustering techniques can be found by Wiggerts (1997).

In a significantly smaller degree there has been used also association rules on program comprehension. Tjortjis, Sinos and Layzell (2003) have proposed a technique that inputs data extracted from source code and derives association rules. These rules help the formation of groups of source code containing interrelated entities. Similar work to Mancoridis et al. (1998) but with the use of association rules instead of clustering techniques has been made by De Oca and Carver (1998). In this work by using the ISA (Identification of Subsystems based on Associations) methodology a software system was decomposed into data cohesive subsystems by mining association rules. Use of association rules into program comprehension has been made also by Sartipi, Kontogiannis and Mavaddat (2000) for architectural design recovery.

PROBLEMS FACED

Bellow we give possible solutions to problems suffered by some current methods by borrowing and using techniques from other fields and mainly association rules.

Surpassing the Low Frequency Problem

As noted by Tjortzis, Sinos and Layzel (2003), input tables produced for source code analysis tend to be sparse. Practically these tables tend to be as sparse as those created by using web pages as input, but they are comparably many orders of magnitude smaller than those from the web. According to Tjortzis, Sinos and Layzel (2003) if one wants to retrieve and incorporate more information about block groups of code he would have to use a small support value. But this would create the so called rare itemset dilemma introduced by Mannila (1998). According to this problem trying to overcome the difficulty of having widely varied itemset frequencies or trying to include rare but very important itemsets (whatever this importance might represent) by lowering the single support measure would result in an overall problematic and highly questionable process. This can be seen better through an example: Suppose we have the case where some itemsets hardly appear in our database, but cannot nevertheless be considered useless (e.g. a variable used to point to an important log file or to alert the user about a serious security exception etc.). On the other hand let there be itemsets that appear very frequently in the data, at a very high percentage. How can one include those infrequent itemsets with the frequent ones, especially if overall the frequencies vary a lot? One is then confronted with the following dilemma: If the overall minsup is set too high, then we eliminate all those itemsets that are infrequent. On the other hand if we set the overall minsup too low, in order to find also those itemsets that are infrequent, we will most certainly generate a huge number of frequent itemsets and will produce far too many rules, rules that will be meaningless. Actually we might end up having generated so many rules that we might have to apply data mining techniques upon the rules themselves. Also apart from generating too many useless rules there exists also the possibility that the whole process will require too much time or will not even finish. As an example the Cover Type dataset[1] from UCI repository with only 120 1-itemsets, but with most of them appearing almost all the time, results in about 15 million large itemsets. Using an associations rules algorithm like Apriori (Agrawal and Srikant, 1994) just to find all these large itemsets required more than 96 CPU hours (Webb, 2000).

In order to overcome this problem and to actually incorporate only the information needed a practicable solution would be to use algorithm MSApriori (Liu, Hsu & Ma, 1999) that uses multiple minimum support values for the items in the database. According to the specific algorithm an important item will receive a low support value whereas an unimportant a higher one. So every itemset including an important one will also receive a low support value, thus finally being discovered and presented to the user as a frequent itemset. That way we can finally include only the itemsets that are actually important to us and not every itemset. The only problem that still remains with that technique is the actual determination of the final support value assigned to every itemset. In contrast to the assumption upon which all association rules approaches work including algorithm MSApriori, that is that the correct support value is already known, in practice the correct minimum support value is not known and can only be estimated based on domain knowledge or on previous experience. Also when talking about the "correct" value, this is judged after the completion of the mining process based on the number of discovered frequent itemsets (i.e. too few or too many itemsets as compared to what has been anticipated), or in other words through completely subjective and rather trivial criteria. In practice algorithm MSApriori can be used only with very few items and only with expert knowledge of the field and the data studied.

The specific problem was addressed in (Kouris, Makris & Tsakalidis, 2004), by using an efficient weighting scheme for assigning the correct minimum support values to all the items. The intuition was simple. Every item can have support at maximum:

$$\max \sup(i) = \frac{T_i}{T_{total}}.$$

The closest the support of an item is set to its maximum support, the more appearances there are needed in order for it to quantify as a frequent and the less important it is considered. So we make a pass over the data without having assigned any support values, and at the end of this pass we know the maximum support of every item. So the final support of an item is determined by

multiplying the maximum support an item can have, with a weight that reflects its importance according to the following formula: sup(i) = wf *max sup(i). The determination of this weight - wf is done by the final user and depends on the application and the data. It remains only for a domain expert to adjust this formula for the case of source code.

Incorporating Additional Information into Data Mining

A big problem suffered by traditional association rules methods was that they treated itemsets and extracted patterns as mere statistical probabilities and as Boolean variables. They did not take into account the exact number of appearances of the items in every transaction and also considered all items of the same importance, thus resulting in their homogenization. This problem is common also in program comprehension. According to Tjortzis, Sinos and Layzel (2003), it would desirable to have a quantitative model instead of a pure qualitative one. In other words we would like to know apart from the existence or absence of a code item in a block of code, also the number of occurrences inside blocks. For example a rule of the form:

$$I_A[3] \rightarrow I_B[2 - 5]$$

would mean "if item A occurs 3 times then item B occurs 2 up to 5 times". We extend this idea and suggest a more complete model that would also consider the actual significance a code item carries. So apart from the exact number of appearances of a code item, we should take into consideration the fact that not all items carry the same significance. An item describing a critical procedure or one linking to many other items is much more significant than one that simply prints out a welcome message. So a different degree of importance should be assigned to every item.

Kouris, Makris and Tsakalidis (2005) have taken these considerations into account for the case of retail data and proposed a complete system. The specific system can be used with minimum modifications for the purpose of program comprehension. The heart of this system is an index created using all past transactions, a mechanism widely used in Information Retrieval for organizing and processing effectively large collections of data and mostly text or web documents (Witten, Moffat & Bell, 1999).

The generated index has various functionalities. In our case a very useful function would be its use as a search engine, by giving answers to queries involving any Boolean operator as well as their combinations. For example how many times does a variable appear with another one, or in which procedures or functions some variables co-appear, or in which blocks specific items appear but never some other ones etc. All these queries would be processed with minimum resources. Also apart from the rule proposed above, another form of a rule of the new system could present would be:

$$I_A \rightarrow I_B, I_E, I_N.$$

where we point to the user with which items item A appears together based though not only on their co-appearance but also on their overall presence. In other words if item A appears 10 times together with item E and 9 times with item B but item E appears totally 100 times whereas item B only 15 then we can easily understand that A is much more correlated with B than with E.

Significant would be also the utilization of the system by making use of ranked queries. When using an index for making ranked queries a model called vector space model is employed, where blocks of code are represented vectors in a multidimensional Euclidean space. Each axis in this space corresponds to code item. The functioning of a vector space model can be divided in three logical phases. The first logical phase is the generation of an index based on our collection of code items, whatever these items might represent. The second is the assignment of a weight to all code items in the index and the third and final phase consists of using some similarity measure to rank the blocks of code that constitute the answer to a user query. So for every code item that interests us, we make a ranked query to the index that tries to match it against the existing ones and propose the user with the most relevant blocks of code. The ranked queries technique and the indexes have been used with great success both in Information Retrieval as well as for data mining (Kouris, Makris & Tsakalidis, 2004), and are expected to present very good results in program comprehension too.

FUTURE TRENDS

As noted above program maintenance and especially program comprehension is the most difficult and costly task in the software lifecycle. In the US the cost of software maintenance has been estimated at more than 70 billion annually for ten billion lines of existing code (Sutherland, 1995), in the UK head officials of UK's TaskForce 2000 estimated the cost of Y2K[2] reprogramming at roughly 50 billion US dollars and at for the same cause Nokia Inc. allocated about 90 million US dollars (Koskinen, 2003). As programs unavoidably become more and more complicated and bulky the specific problem will rather deteriorate. In our opinion future research efforts will concentrate on adopting and adapting techniques from other fields (such as data mining and information retrieval) in order to solve specific problems in the field.

CONCLUSION

Program comprehension is a critical task in the software life cycle. In this work we addressed an emerging field, namely program comprehension through data mining. The specific task is considered by many researchers to be one of the "hottest" ones nowadays, with large financial and research interest. As we made apparent in this paper, the use of data mining techniques and especially association rules can facilitate better and easier maintenance of software and extend its lifetime.

REFERENCES

Agrawal, R., & Srikant, R. (1994). Fast Algorithms for Mining Generalized Association Rules, In *Proccedings of 20th VLDB Conference*, Santiago, Chile, September, 1994. 487-499.

Anquetil, N., & Lethbridge, T. C. (1999). Experiments with Clustering as a Software Remodularization method. In *Proceedings of 6th Working Conference Reverse Engineering (WCRE 99)*, Atlanta, Georgia, USA, October, 1999. 235-255.

Chen, M.S., Han, J. & Yu, P.S., (1996). Data Mining: An Overview from a Database Perspective. *IEEE Transactions on Knowledge and Data Engineering, 8*(6), 866-883.

De Oca, C.M., & Carver, D.L. (1998). Identification of Data Cohesive Subsystems Using Data Mining Techniques. In *Proceedings of International Conference on Software Maintenance*, Bethesda, Maryland, November, 1998. 16-23.

Erdil, K. ,Finn, E., Keating, K, Meattle, J., Park, S. & Yoon, D. (2003). Software Maintenance as Part of the Software Life Cycle. Dept. of Computer Science, Tufts University, Retrieved February 14, 2006, from http://hepguru.com/maintenance/Final_121603_v6.pdf

Erdös, K. & Sneed, H., M. (1998). Partial Comprehension of Complex Programs (enough to perform maintenance). In *Proceedings of 6th International Workshop on Program Comprehension*, Ischia, Italy, June, 1998. 98-105.

Kanellopoulos, Y., & Tjortjis, C. (2004). Data Mining Source Code to Facilitate Program Comprehension: Experiments on Clustering Data Retrieved from C++ Programs. In *Proceedings of 12th IEEE International Workshop on Program Comprehension*, Bari, Italy, June 24-26, 2004. 214-226.

Koskinen, J. (2003). Software Maintenance Cost. Retrieved February 18, 2006, from http://www.cs.jyu.fi/~koskinen/smcosts.htm

Kouris, I.N., Makris C.H., & Tsakalidis, A.K. (2005). Using Information Retrieval techniques for supporting data mining. *Data & Knowledge Engineering, 52*(3), 353-383.

Kouris, I.N., Makris, C.H., & Tsakalidis, A.K. (2004). Assessing the Microeconomic Facet of Association Rules via an Efficient Weighting Scheme. In *Proceedings of ICKM 2004*, Singapore, December 13-15, 2004. 13-15.

Lakhotia, A. (1997). A Unified Framework for Expressing Software Subsystem Classification Techniques. *Journal of Systems and Software, 36*(3), 211-231.

Liu, B., Hsu, W., & Ma, Y. (1999). Mining Association Rules with Multiple Minimum Supports. In *Proceedings of ACM KDD Conference*, San Diego, CA, USA, August, 1999. 337-341.

Mancoridis, S., Mitchell, B.S., Rorres, C., Chen, Y. & Gansner, E.R. (1998). Using Automatic Clustering to

Produce High-Level System Organizations of Source Code. In *Proceedings of 6th Int'l Workshop Program Understanding,* Ischia, Italy, June, 1998. 45-53.

Mannila, H. (1998). Database methods for Data Mining. *Tutorial presented at the 4th ACM KDD Conference, August 24-27, New York, USA, 1998.*

Pigoski, T.M. (1997). *Practical Software Maintenance: Best Practices for Managing your Software Investment.* New York: Wiley Computer Publishing.

Sartipi, K., Kontogiannis, K., & Mavaddat, F. (2000). Architectural Design Recovery Using Data Mining Techniques. In *Proceedings of 2nd European Working Conf. Software Maintenance Reengineering,* Zurich, Switzerland, February, 2000. 129-140.

Sommerville, I. (2001). *Software Engineering.* (6th ed.). Harlow: Addison-Wesley.

Sutherland, J. (1995). Business Objects in Corporate Information Systems. *ACM Computing Surveys 27*(2), 274-276.

Tjortjis, C. & Layzel, P. J. (2001). Using Data Mining to Assess Software Reliability. In *Suppl. Proc. of IEEE 12th International Symposium on Software Reliability Engineering,* Hong Kong, November 27-30, 2001. 221-223.

Tjortjis, C., Sinos, L. & Layzell P. (2003). Facilitating program comprehension by mining association rules from source code. In *Proceedings of IEEE 11th Workshop Program Comprehension,* Portland, Oregon, USA, May 10-11, 2003. 125-132.

Tzerpos, V. & Holt, R. (1998). Software Botryology: Automatic Clustering of Software Systems. In *Proceedings of 9th International Workshop on Database Expert Systems Applications,* Vienna, Austria, August 24-28, 1998. 811-818.

Van Vliet, H. (2000). *Software Engineering: Principles and Practices.* (2nd ed.). West Sussex, England: John Wiley & Sons.

Von Mayrhauser, A.& Vans, A. M. (1994). Program Understanding – A Survey. (Technical Report CS-94-120). Department of Computer Science, Collorado State University.

Webb, G. I. (2000). Efficient search for association rules. In *Proceedings of 6th ACM KDD Conference,* Boston, Massachusetts, USA, August 20-23. 99-107.

Wiggerts, T.A. (1997). Using Clustering Algorithms in Legacy Systems Remodularization. In *Proceedings of 4th Working Conference on Reverse Engineering,* Amsterdam, The Netherlands, October 6-8, 1997. 33-43.

Witten, I., Moffat, A., & Bell, T. (1999). *Managing Gigabytes: Compressing and Indexing Documents and Images.* (2nd ed.). San Francisco: Morgan Kaufmann.

Zelkowitz, M. V., Shaw, A. C., & Gannon, J. D. (1979). *Principles of Software Engineering and Design.* Englewood Cliffs, NJ: Prentice-Hall.

KEY TERMS

Software Maintenance: Process involved with enhancing and optimizing deployed software (e.g by the addition of new functionalities), updating previous versions, as well as remedying defects and deficiencies found after deployment (such as bugs).

Program Comprehension: The task dealing with the processes (cognitive or others) used by software engineers to understand programs.

Software Life Cycle: The phases a software product goes through from its initial conception until it is finally rendered obsolete. The continuing process, in which a software project specifies, prototypes, designs, implements, tests, and maintains a piece of software.

Data Mining: Analysis of data in a database using tools which look for trends or anomalies without knowledge of the meaning of the data. The nontrivial extraction of implicit, previously unknown, and potentially useful information from data. The science of extracting useful information from large data sets or databases.

Data Clustering: Clustering is the unsupervised classification of patterns (observations, data items, or feature vectors) into groups (clusters).

Association Rules for Program Comprehension: Finding frequent patterns, associations, correlations, or

causal structures among sets of data items extracted directly from source code. These rules aim at capturing program structure and thus achieving better system understanding.

Itetmsets: A collection or a combination of items (such as variables in the source code of a program) in a database.

ENDNOTES

1 Available at http://kdd.ics.uci.edu/. A test dataset collected for the purpose of predicting forest cover type from cartographic variables only, used like all other datasets in this collection in order to test algorithms to large and complex datasets.

2 Year 2000 problem also known millennium bug, was a bug in program design that caused some date-related processing to operate incorrectly for dates and times on and after January 1, 2000.

Program Mining Augmented with Empirical Properties

Minh Ngoc Ngo
Nanyang Technological University, Singapore

Hee Beng Kuan Tan
Nanyang Technological University, Singapore

INTRODUCTION

Due to the need to reengineer and migrating aging software and legacy systems, reverse engineering has started to receive some attention. It has now been established as an area in software engineering to understand the software structure, to recover or extract design and features from programs mainly from source code. The inference of design and feature from codes has close similarity with data mining that extracts and infers information from data. In view of their similarity, reverse engineering from program codes can be called as program mining. Traditionally, the latter has been mainly based on invariant properties and heuristics rules. Recently, empirical properties have been introduced to augment the existing methods. This article summarizes some of the work in this area.

BACKGROUND

"Software must evolve over time or it becomes useless" (Lehman). A large part of software engineering effort today is involved not in producing code from scratch but rather in maintaining and building upon existing code. For business, much of their applications and data reside in large, legacy systems. Unfortunately, these systems are poorly documented. Typically, they become more complex and difficult to understand over time.

Due to this need to reengineer and migrating aging software and *legacy systems*, reverse engineering has started to receive some attentions. *Reverse engineering* is an approach to understand the software structure, to recover or extract design and features, given the source code. This process identifies software building blocks, extract structural dependencies, produces higher-level abstractions and present pertinent summaries. Reverse engineering helps by providing computer assistance, often using compiler technologies such as lexical, syntactic and semantic analysis. *Static analysis* strengthens the study by inferring relationships that may not be obvious from the syntax of the code, without running the program.

The inference of *design patterns* and features from codes has close similarity with data mining that extracts and infers information from data. In view of their similarity, reverse engineering from program codes can be called as program mining. Program mining is a challenging task because there are intrinsic difficulties in performing the mapping between the language of high level design requirements and the details of love level implementation. Although program mining depends heavily on human effort, a number of approaches have been proposed to automate or partially automate this process.

Traditionally, approaches to program mining are based on invariant properties of programs or heuristic rules to recover design information from source code (De Lucia, Deufemia, Gravino, & Risi, 2007; Poshyvanyk, Gueheneuc, Marcus, Antoniol, & Rajlich, 2007; Robillard & Murphy, 2007; Shepherd, Fry, Hill, Pollock, & Vijay-Shanker, 2007; Shi & Olsson, 2006). Recently, several researchers have developed *experimental program analysis* approaches (Ruthruff, Elbaum, & Rothermel, 2006; Tonella, Torchiano, Du Bois, & Systa, 2007) as a new paradigm for solving software engineering problems where traditional approaches have not succeeded. The use of empirical properties, which have been validated statistically, to solve problems is very common in the area of medicine (Basili, 1996). However, the application of this method is rather unexplored in software engineering. This paper summarizes some of our work in this area which incorporates the use of empirical properties.

MAIN FOCUS

In this section, we first describe our work on empirical-based recovery and maintenance of input error correction *features* in information system (Ngo & Tan, 2006).We then discuss the use of empirical properties to infer the infeasibility of a program path (Ngo & Tan, 2007). This work on infeasible path detection is useful for all software engineering tasks which rely on static analysis especially testing and coverage analysis. Finally, we present an approach to extract all the possible database interactions from source code (Ngo, Tan, & Trinh, 2006).

Empirical Recovery and Maintenance of Input Error Correction Features

Information systems constitute one of the largest and most important software domains in the world. In many information systems, a major and important component is processing inputs submitted from its external environment to update its database. However, many input errors are only detected after the completion of execution. We refer to this type of input errors as after-effect input errors. As after-effect input error is unavoidable, the provision of after-effect input error correction features is extremely important in any information system. Any omission of the provision of these features will lead to serious adverse impact. We observe that input error correction features exhibits some common properties. Through realizing these properties from source code, input error correction features provided by a system to correct these effects can be recovered. All the empirical properties have been validated statistically with samples collected from a wide range of information systems. The validation gives evidence that all the empirical properties hold for more than 99 percent of all the cases at 0.5 level of significance.

Properties of Input Error Correction Features

Input errors can be approximately classified into: error of commission (EC), error of omission (EO) and value error (VE). In a program, there are statements which when being executed will raise some external effects; these statements are called effect statements. If effect statements are influenced by some input errors, they will result in erroneous effects; we refer to these as effect errors. Effect errors can be classified into EO, EC and VE in the combination of attributes for executing the effect statement.

In an information system, a type of effect error (EO, EC or VE) that may occur in executing an effect statement e can be corrected by executing another effect statement f in the system. We call f an error correction statement (Tan & Thein, 2004) for correcting e. The minimum collection of all the paths for correcting an input error ξ is called the basis collection of error correction paths for correcting ξ. We discover some empirical patterns for realizing the basis collection of error correction paths for correcting an input error as follow:

Empirical Property 1. A set of paths $\{q_1,\ldots, q_k\}$ is a basis collection of error correction paths for correcting the input ξ is and only if we can partition the set of effect errors resulting from ξ into several partitions $\{E_1,\ldots., E_k\}$ such that for each j, $1 \leq j \leq k$, by executing q_j, all the effect errors in E_j are corrected.

For many programs, the correctability of all its input errors can be deduced from existence of basis collection of error correction paths for some of these errors. This is presented in the following empirical property:

Empirical Property 2. If for each path through the control flow graph of a program, there is a basis collection of error correction paths for correcting EC in the input accessed in the path, then any after-effect input error of the program is correctable.

If each path through the control flow graph of a program S is in a basis collection of error correction paths for correcting an input error of program T, then S is called an error correction program for T. The following empirical property suggests a mechanism to verify the correctability of a program V.

Empirical Property 3. It is highly probable that any input error of program V is correctable if and only if one of the following conditions holds:

- Based on basis collections of error correction paths, Empirical Property 2 infers that any after-effect error of V is correctable.
- Program V is an error correction program for program T and any after-effect error of T is correctable.

Automated Recovery of the Input Error Correction Features

Based on the empirical properties established input error correction features provided by a program can be automatically recovered through the following three steps:

- **Step 1—Program analysis:** Each program in the system is transformed into a control flow graph. For each program, the set of all the input errors and resulting effect errors are derived.
- **Step 2—Recovery of input correction features:** For each basic type ξ of input errors, Empirical Property 1 is applied to compute a basis collection of error correction paths for correcting the input error.
- **Step 3—Recovery of program input error correctability:** In this step, Empirical Property 2 and 3 are applied to recover a set of programs, whose input errors are correctable.

Maintaining Input Error Correction Features

We apply the decomposition slicing technique introduced by (Gallagher & Lyle, 1991) to decompose a program with respect to the input error correction features. By doing this, we are able to isolate program statements which implement an error correction feature. We also obtain a smaller view of a correction program with the unrelated statements removed and dependent statements restricted from modification. To apply this technique, we introduce the concept of effect-oriented decomposition slice. The idea is that an effect-oriented decomposition slice of a program, taken with respect to the external effects raised during the execution of the program, should contain only statements which affect the attributes of some external effects raised during the execution of the program.

To maintain a correction set for correcting a given input error ξ, first of all, one needs to compute both the decomposition slice of the effect errors caused by ξ and all the decomposition slices of the correction paths for correcting ξ. After having these computed, removing and adding an error correction feature can be done easily by following the four rules presented by (Gallagher & Lyle, 1991).

Infeasible Path Detection

A great majority of program paths are found to be infeasible, which in turn make static analysis overly conservative. As static analysis plays a central part in many software engineering activities, knowledge about infeasible program paths can be used to greatly improve the performance of these activities especially structural testing and coverage analysis. We have proposed an empirical approach to the problem of infeasible path detection (Ngo & Tan, 2007). We discovered that many infeasible paths exhibit some common properties which are caused by four code patterns including identical/complement-decision, mutually-exclusive-decision, check-then-do and looping-by-flag pattern. Through realizing these properties from source code, many infeasible paths can be precisely detected. Binomial tests have been conducted which give strong statistical evidences to support the validity of the empirical properties. Our experimental results show that even with some limitations in the current prototype tool, the proposed approach accurately detects 82.3% of all the infeasible paths.

Next, we shall characterize four code patterns and the type of infeasible paths caused by each one of them. Our empirical study shows that infeasible paths detected in these code patterns constitute a very large proportion of infeasible paths.

Identical/Complement-Decision Pattern

We have discovered that, sometimes, actions to be performed under the same or completely different conditions could be implemented by separate independent selection constructs (if statements) with identical or complement condition. This is referred to as identical/complement-decision pattern.

Let p and q be two if-statements in a program. If p and q follow identical/complement-decision pattern then any path that contains branches of p and q with syntactically different branch predicates is infeasible.

Mutually-Exclusive-Decision Pattern

Empirically, we have discovered that a decision based on the same set of factors for one purpose is often implemented in a program using two approaches. One alternative for implementing this type of decision is

to use nested-if structure: the next condition will only be checked if the previous condition fails. The second alternative avoids the use of nested-if structure. It formulates a set of mutually exclusive conditions based on the set of factors for making the decision so that a separate selection construct for each condition with only one non-null branch is used to jointly implement the decision. This design pattern is referred to as mutually-exclusive-decision pattern.

Though from code optimization and infeasible path avoidance viewpoint, the first alternative is clearly better, from our empirical observation, the mutually-exclusive decision pattern (second alternative) is frequently used. We believe this could be due to the latter alternative provides a neater code structure.

Any path that contains more than one true branch of mutually-exclusive-decision conditions is infeasible. This is because of the uniqueness of decision which means at most one decision is made at one point in time.

Check-Then-Do Pattern

In many situations, a set of actions is only performed upon the successful checking of a set of conditions. The commonly used design method to implement the above-mentioned requirement is to separate the checking of conditions from the actions to be performed. The checking of conditions reports the outcome through setting a 'flag' variable. Actions are performed in a branch of a predicate node which tests the value of the 'flag' variable. This design pattern is referred to as check-then-do pattern. In this code pattern, any path that fails the checking of the condition and still goes trough the statement which executes the action is feasible. Likewise any path that passes the checking of the condition but does not go through the execution of the action is also infeasible.

Looping-by-Flag Pattern

Sometimes, programmers use a 'flag' variable to control the termination of a loop. Normally, the flag variable is initialized to a value which enables the execution of the loop. Inside the body of the loop, the flag variable will be set to another value if some conditions are satisfied. This value of the flag variable will lead to the termination of the loop. This coding pattern is often referred to as looping-by-flag pattern.

Assuming that p is a path that contains statements following looping-by-flag pattern. If p enters the statement which sets the flag variable to a value which lead to the termination of the loop and then re-enters the loop is infeasible.

Automated Extraction of Database Interactions

Database interactions are among the most essential functional features in web applications. Thus, for the maintenance and understanding of web applications, it is vital that the web engineer could identify all code segments which implement the database interactions features. Unfortunately, the highly dynamic nature of web applications makes it challenging to automatically extract all the possible database interactions from source code. Recent work on extracting functionalities from source code (De Lucia et al., 2007; Poshyvanyk et al., 2007; Robillard & Murphy, 2007; Shepherd et al., 2007) do not differentiate between different types of interactions. Moreover, to the best of our knowledge, no approach to extracting functionalities from source code is capable of dealing with the dynamic nature of web applications.

To overcome this problem, we propose an approach using symbolic execution and inference rules to extract all the possible database interactions automatically (Ngo, Tan & Trinh, 2006). These rules are basically empirical properties of the database interactions. The basic idea behind our approach is that we identify all the paths which could lead to a database interaction and symbolically execute the path by applying the proposed symbolic evaluation rules. We also develop Inference rules, which can be used to infer all the possible database interactions from symbolic expressions. Importantly, our contributions have several advantages, namely: it can extract all the possible interactions; it can distinguish between different types of interactions; it also generates the execution trace to a particular database interaction, which is useful for understanding and maintenance of web applications.

We have introduced Interaction Flow Graph (IFG) which facilitates the extraction of database interactions without having to consider other unrelated program statements. Our approach has four main steps. In the first step, all interaction paths are identified. In the second step, a slice of the interaction path is calculated which contains all the statements involving in the interactions.

In the third step, the set of symbolic rules are applied to symbolically execute each slice. In the last step, inference rules are used to infer all the possible interaction types from each symbolic expression derived in the third step.

The approach was examined via extensive experiments on four open-source web applications. We manually measured the precision of the approach and concluded that the approach precisely extracts all the possible database interactions. Our approach shows usefulness in the testing during system development and maintenance of web applications which involve extensive database interactions.

FUTURE TRENDS

The use of empirical results provide a simple yet efficient way to many software engineering tasks which rely on program analysis, especially program-mining. Experimental program analysis might be able to draw inferences about the properties of a software system in which traditional analyses have not succeeded.

Indeed, many characteristics of scientific experimentation have been utilized by program analysis techniques such as formalization, hypothesis testing or the use of sample cost-effectively explore effects to large population in generalizable manners. Take for example the use of empirical techniques in detecting infeasible paths that we have discussed in Section 3.2. We first routinely form hypotheses about patterns of infeasible program path. We then collect random sample from various application domains. From the results of hypothesis testing, we can either accept or reject the null hypotheses. From here, conclusions can be drawn or new hypothesis can be created. This process is repeated until we can conclude that our hypotheses hold for a high percentage (for e.g. 99 percent) of all the cases in the sample. The similar approach could be found in existing program analysis techniques such as features localization or design pattern recovery (De Lucia et al., 2007; Heuzeroth, Mandel, & Lowe, 2003; Poshyvanyk et al., 2007; Robillard & Murphy, 2007; Shepherd et al., 2007; Shi & Olsson, 2006; Zhang, Young, & Lasseter, 2004). These approaches are inherently experimental in nature.

Experimental program analysis is undoubtedly the future trends for many program analysis tasks. Ruthruff et al. (2006) have concluded on this future trend as follow: *"We believe that experimental program analysis provides numerous opportunities for program analysis and software engineering research. We believe that it offers distinct advantages over other forms of analysis at least for particular classes of analysis tasks, including procedures for systematically controlling sources of variation in order to experimentally analyze software systems, and experimental designs and sampling techniques that reduce the cost of generalizing targeted aspects of a program. We believe that such advantages will lead to significant advances in program analysis research and in the associated software engineering technologies that this research intends to improve."*

CONCLUSION

In this article, we have discussed the use of empirical property to facilitate program mining. We have summarized our major work in this area including an approach to recover and maintain the input error correction features provided by an information system; an approach to detect infeasible path through recognizing their most common patterns; and an approach to extract all the database interactions in a web applications through using empirical symbolic and inference rules.

All the approaches are based on empirical properties and program analysis techniques. However, there is a major theoretical difference between the proposed approach and other approaches that uses program analysis for program mining. The proposed approach is developed through the integration of invariant and empirical properties. All the empirical properties have been validated statistically with samples collected from a wide range of application domains.

Based on the empirical properties, our approach automatically recovers various types of information from source code. The novelty of our approach lies in the use of empirical properties, which provides a simple yet efficient approach to program mining. We believe that this is a promising future direction, which opens a new avenue for the automated reverse engineering.

REFERENCES

Basili, V. R. (1996). *The role of experimentation in software engineering: past, current, and future.* Paper presented at the Proceedings of IEEE 18th International

Conference on Software Engineering, 25-30 March 1996, Berlin, Germany.

De Lucia, A., Deufemia, V., Gravino, C., & Risi, M. (2007). *A two phase approach to design pattern recovery.* Paper presented at the 2007 11th European Conference on Software Maintenance and Reengineering, 21-23 March 2007, Amsterdam, Netherlands.

Gallagher, K. B., & Lyle, J. R. (1991). Using program slicing in software maintenance. *IEEE Transactions on Software Engineering, 17*(8), 751-761.

Heuzeroth, D., Mandel, S., & Lowe, W. (2003). *Generating design pattern detectors from pattern specifications.* Paper presented at the Proceedings 18th IEEE International Conference on Automated Software Engineering, 6-10 Oct. 2003, Montreal, Que., Canada.

Lehman, M. M. Programs, life cycles, and laws of software evolution. *Proceedings of the IEEE, 68*(9), 1060-1076.

Ngo, M. N., & Tan, H. B. K. (2007, September 2007). *Detecting large number of infeasible paths through recognizing their patterns.* Paper presented at the Proceedings of the the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering, Dubrovnik, Croatia.

Ngo, M. N., Tan, H. B. K., & Trinh, D. (2006). *Automated Extraction of database interactions in web applications.* Paper presented at the 14th IEEE International Conference on Program Comprehension, Athens, Greece.

Poshyvanyk, D., Gueheneuc, Y.-G., Marcus, A., Antoniol, G., & Rajlich, V. (2007). Feature location using probabilistic ranking of methods based on execution scenarios and information retrieval. *IEEE Transactions on Software Engineering, 33*(6), 420-432.

Robillard, M. P., & Murphy, G. C. (2007). Representing concerns in source code. *ACM Transactions on Software Engineering and Methodology, 16*(1), 3.

Ruthruff, J. R., Elbaum, A. S., & Rothermel, A. G. (2006). *Experimental program analysis: a new program analysis paradigm.* Paper presented at the Proceedings of the 2006 international symposium on Software testing and analysis, Portland, Maine, USA.

Shepherd, D., Fry, Z. P., Hill, E., Pollock, L., & Vijay-Shanker, K. (2007). *Using natural language program analysis to locate and understand action-oriented concerns.* Paper presented at the 6th International Conference on Aspect-Oriented Software Development, Mar 12-16 2007, Vancouver, BC, Canada.

Shi, N., & Olsson, R. A. (2006). *Reverse engineering of design patterns from Java source code.* Paper presented at the Proceedings. 21st IEEE International Conference on Automated Software Engineering, 18-22 Sept. 2006, Tokyo, Japan.

Tan, H. B. K., & Thein, N. L. (2004). Recovery of PTUIE handling from source codes through recognizing its probable properties. *IEEE Transactions on Knowledge and Data Engineering, 16*(10), 1217-1231.

Tonella, P., Torchiano, M., Du Bois, B., & Systa, T. (2007). Empirical studies in reverse engineering: State of the art and future trends. *Empirical Software Engineering, 12*(5), 551-571.

Zhang, X., Young, M., & Lasseter, J. H. E. F. (2004). *Refining code-design mapping with flow analysis.* Paper presented at the Twelfth ACM SIGSOFT International Symposium on the Foundations of Software Engineering, SIGSOFT 2004/FSE-12, Oct 31-Nov 5 2004, Newport Beach, CA, United States.

KEY TERMS

After-Effect Input Error: An after-effect input error is a user input error which is only discovered after the execution of the program.

Control Flow Graph: A control flow graph contains a set of nodes and a set of edges in which, a node represents a program statement and an edge represents the transfer of control between statements.

Database Interaction: A node in the control flow graph of a program which executes, in an arbitrary run of the program, one of the SQL data manipulation language operations select, insert, update or delete is defined as an interaction node. A database interaction in a program *P* corresponds to an execution of an interaction node in the control flow graph of *P*.

Decomposition Slicing: A decomposition slice with respect to a variable *v* captures all the computation of *v* and is independent of program location.

Infeasible Path: An infeasible path is a program path for which there is not input such that the path can be executed.

Program-Mining: In software engineering, program-mining refers to reverse engineering which is a process of identifying software components, their inter-relationships and representing these entities at a higher level of abstraction.

Program Slicing: A slice S(v, n) (of program P) on variable v, or a set of variables, at statement n yields the portions of the program that contributed to the value v of just before statement n is executed.

Projected Clustering for Biological Data Analysis

Ping Deng
University of Illinois at Springfield, USA

Qingkai Ma
Utica College, USA

Weili Wu
The University of Texas at Dallas, USA

INTRODUCTION

Clustering can be considered as the most important unsupervised learning problem. It has been discussed thoroughly by both statistics and database communities due to its numerous applications in problems such as classification, machine learning, and data mining. A summary of clustering techniques can be found in (Berkhin, 2002).

Most known clustering algorithms such as DBSCAN (Easter, Kriegel, Sander, & Xu, 1996) and CURE (Guha, Rastogi, & Shim, 1998) cluster data points based on full dimensions. When the dimensional space grows higher, the above algorithms lose their efficiency and accuracy because of the so-called "curse of dimensionality". It is shown in (Beyer, Goldstein, Ramakrishnan, & Shaft, 1999) that computing the distance based on full dimensions is not meaningful in high dimensional space since the distance of a point to its nearest neighbor approaches the distance to its farthest neighbor as dimensionality increases. Actually, natural clusters might exist in subspaces. Data points in different clusters may be correlated with respect to different subsets of dimensions. In order to solve this problem, feature selection (Kohavi & Sommerfield, 1995) and dimension reduction (Raymer, Punch, Goodman, Kuhn, & Jain, 2000) have been proposed to find the closely correlated dimensions for all the data and the clusters in such dimensions. Although both methods reduce the dimensionality of the space before clustering, the case where clusters may exist in different subspaces of full dimensions is not handled well.

Projected clustering has been proposed recently to effectively deal with high dimensionalities. Finding clusters and their relevant dimensions are the objectives of projected clustering algorithms. Instead of projecting the entire dataset on the same subspace, projected clustering focuses on finding specific projection for each cluster such that the similarity is reserved as much as possible.

BACKGROUND

Projected clustering algorithms generally fall into two categories: density-based algorithms (Agrawal, Gehrke, Gunopulos, & Raghavan, 1998; Procopiuc, Jones, Agarwal, & Murali, 2002; Liu & Mamoulis, 2005; Ng, Fu, & Wong, 2005; Moise, Sander, & Ester, 2006) and distance-based algorithms (Aggarwal, Procopiuc, Wolf, Yu, & Park, 1999; Aggarwal & Yu, 2000; Deng, Wu, Huang, & Zhang, 2006; Yip, Cheung, & Ng, 2003; Tang, Xiong, Zhong, & Wu, 2007). Density-based algorithms define a cluster as a region that has a higher density of data points than its surrounding regions. Dense regions only in their corresponding subspaces need to be considered in terms of projected clustering. Distance-based algorithms define a cluster as a partition such that the distance between objects within the same cluster is minimized and the distance between objects from different clusters is maximized. A distance measure is defined between data points. Compared to density-based methods in which each data point is assigned to all clusters with a different probability, distance-based methods assign data to a cluster with probability 0 or 1. Three criteria (Yip, Cheung, & Ng, 2003) have been proposed to evaluate clusters: the number of data points in a cluster, the number of selected dimensions in a cluster, and the distance between points at selected dimensions.

PROCLUS (Aggarwal, Procopiuc, Wolf, Yu, & Park, 1999) is a typical distance-based projected clustering algorithm which returns a partition of the data points, together with sets of dimensions on which data points in each cluster are correlated by using Manhattan segmental distance. However, this algorithm loses its effectiveness when points in different dimensions have different variance. We propose our algorithm, IPROCLUS (Improved PROCLUS) based on the following enhancements. We propose the modified Manhattan segmental distance which is more accurate and meaningful in projected clustering in that the closeness of points in different dimensions not only depends on the distance between them, but also relates to the distributions of points along those dimensions. Since PROCLUS strongly depends on two user parameters, we propose a dimension tuning process to reduce the dependence on one of the user parameters. We also propose a simplified replacing logic compared to PROCLUS.

MAIN FOCUS

Our algorithm, IPROCLUS, which allows the selection of different subsets of dimensions for different clusters, is based on PROCLUS. Our algorithm takes the number of clusters k and the average number of dimensions l in a cluster as inputs. It has three phases: an initialization phase, an iterative phase, and a cluster refinement phase. The medoid for a cluster is the nearest data point to the center of the cluster. The detail of our algorithm can be found in (Deng, Wu, Huang, & Zhang, 2006).

Modified Manhattan Segmental Distance

Manhattan segmental distance is defined as $(\sum_{i\in D}|p_{1,i}-p_{2,i}|)/|D|$ in PROCLUS. In our algorithm, we propose the modified Manhattan segmental distance as the distance measure to improve accuracy. We find that the closeness of points in different dimensions not only depends on the distance between them, but also depends on the distributions of points along different dimensions. Therefore we define a normalization factor n_i for each dimension, which is the standard deviation of all points in a dataset along dimension i. The modified Manhattan segmental distance between x_1 and x_2 relative to dimension set D can be defined as: $(\sum_{i\in D}|p_{1,i}-p_{2,i}|/n_i)/|D|$.

Initialization Phase

In the initialization phase, all data points are first chosen by random to form a random data sample set S with size A×k, where A is a constant. Then S is chosen by a greedy algorithm to obtain an even smaller set of points M with size B×k, where B is a small constant. The greedy algorithm (Gonzalez, 1985) is based on avoiding choosing the medoids from the same cluster. Therefore, the set of points which are most far apart are chosen.

Iterative Phase

We begin by choosing a random set of k points from M. Then the bad medoids (Aggarwal, Procopiuc, Wolf, Yu, & Park, 1999) in the current best medoids set are iteratively replaced with random points from M until the current best medoids set does not change after a certain number of replacements have been tried.

In each iteration, we first find dimensions for each medoid in the set, and form the cluster corresponding to each medoid. Then the clustering is evaluated and the bad medoids in the current best medoids set are replaced if the new clustering is better.

In order to find dimensions, several terms need to be defined first. For each medoid m_i, δ_i is the minimum distance from any other medoids to m_i based on full dimensions. The locality L_i is the set of points within the distance of δ_i from m_i. $X_{i,j}$ is the average distance to m_i along dimension j, which is calculated by dividing the average distance from the points in L_i to m_i along dimension j by the normalization factor n_j. There are two constraints when associating dimensions to medoids. The total number of dimensions associated to medoids must be equal to $k\times l$. The number of dimensions associated with each medoid must be at least 2. For each medoid i, we compute the mean $Y_i=\left(\sum_{j=1}^{d}X_{i,j}\right)/d$, and the standard deviation $\sigma_i=\sqrt{\sum_j\left(X_{i,j}-Y_i\right)^2/(d-1)}$ of the values $X_{i,j}$. Y_i represents the average modified Manhattan segmental distance of the points in L_i relative to the entire space. Thus $Z_{i,j}=\left(X_{i,j}-Y_i\right)/\sigma_i$ indicates how the average distance along dimension j associated with the medoid m_i is related to the average

modified Manhattan segmental distance associated with the same medoid. Dimensions for all clusters are decided by picking the smallest $Z_{i,j}$ values by a greedy algorithm (Ibaraki & Katoh, 1988) such that the above two constraints are met.

A single pass is done over the database to assign the points to the medoids. For each *i*, we compute the modified Manhattan segmental distance relative to D_i between a point and the medoid m_i, and assign the point to the closest medoid. The quality of a set of medoids is evaluated by the average modified Manhattan segmental distance from the points to the centroids of the clusters to which they belong.

We propose a simplified replacing logic compared to PROCLUS to decide whether to replace the bad medoids in the current best medoids set with new medoids. When replacing the bad medoids, δ_i for each medoid m_i is calculated first. We only recalculate the $X_{i,j}$ values for those medoids whose δ_i values change (store the $X_{i,j}$ value for current best objective case so that for those medoids whose δ_i values don't change, their $X_{i,j}$ values can be recovered from the stored values). Then we calculate Y_i, σ_i and $Z_{i,j}$. Dimensions are picked for all clusters by the $Z_{i,j}$ values. When we assign points to clusters, there are two cases. For the points previously in the dimensions whose δ values don't change, we only compare their modified Manhattan segmental distance from the current medoids with their modified Manhattan segmental distance from the medoids whose δ values change. For the points previously in the dimensions whose δ values change or in the bad medoid's cluster, we compare its distance to all the current medoids to decide which cluster it belongs to. Then the new clusters are evaluated to decide whether the objective value is better. The simplified logic for assigning points in the iterative phase is achieved by the IterativeAssignPoints function (Deng, Wu, Huang, & Zhang, 2006).

Refinement Phase

In the refinement phase, we redo the process in the iterative phase once by using the data points distributed by the result clusters at the end of the iterative phase, as opposed to the localities of the medoids. Once the new dimensions have been computed, the points are reassigned to the medoids relative to these new sets of dimensions. Outliers are also handled during the last pass over the data as in PROCLUS.

Users need to specify the average number of dimensions denoted as *l* in PROCLUS. Although it has achieved that different clusters have different subsets of dimensions, the number of dimensions for each cluster is still under the control of *l*. According to one of the three criteria discussed in the background section, we want the number of dimensions in a cluster to be as large as possible. Therefore, we propose one more step at the end of the refinement phase to reduce the dependence on *l*. In this step, for each cluster *i*, we choose the dimension with the smallest $Z_{i,j}$ value from the dimensions that are not chosen in previous steps and add it to the dimensional space. If the new cluster is better, we keep the newly added dimension and repeat this process to try to add more dimensions; otherwise, it will be discarded and we stop trying for this cluster. This process is achieved by the DimensionTuning algorithm (Deng, Wu, Huang, & Zhang, 2006). The quality of a cluster is evaluated by the remaining two criteria. A user-defined threshold is set for the number of points. Clusters that pass the threshold will be evaluated by the distance between points. All criteria have been considered in our algorithm, which gives a more balanced result.

Empirical Results and Analysis

The experimental evaluation was performed on a Dell Dimension 4600 Intel Pentium IV processor 2.4GHz with 1.00GB of memory, running Windows XP professional with service pack 2. The data was stored on a 7200RPM, 8MB cache, 80G hard drive. The flow chart of the experimental evaluation for a dataset is illustrated in Figure 1.

We test the performance of IPROCLUS and PROCLUS for synthetic data and real biological data. Synthetic data generation and the results of running PROCLUS and IPROCLUS on synthetic datasets can be found in (Deng, Wu, Huang, & Zhang, 2006). In a summary, our algorithm shows much higher accuracy and lower dependence on one user input than PROCLUS. In addition, two algorithms show comparable running time since the simplified replacing logic offsets the running time increase from the dimension tuning.

One challenge of applying clustering algorithms on gene expression data is the huge number of genes (dimensions) involved. Projected clustering algorithms are designed to deal with high dimensionalities. This leads us to test the projected clustering algorithms on

Figure 1. Flow chart of the experiment

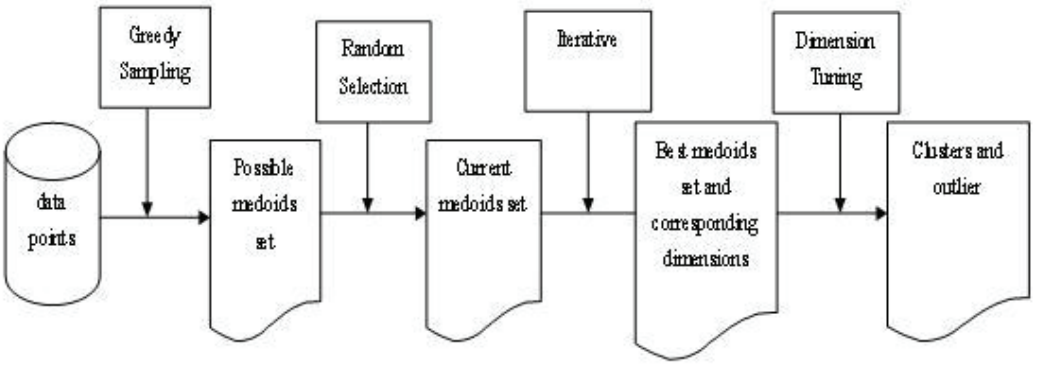


Figure 2. Confusion matrices on the colon tumor dataset

Input / Output	Normal	Tumor
1	18	6
2	4	34

(a) IPROCLUS

Input / Output	Normal	Tumor
1	5	17
2	12	28

(b) PROCLUS

biological datasets. We compared the performance of IPROCLUS and PROCLUS on the Colon Tumor dataset (Alon et al., 1999) and the Leukemia dataset (Golub et al., 1999). These two biological datasets are well documented and have been widely used in the bioinformatics research.

The Colon Tumor dataset consists of the expression values on 2000 genes of 40 tumor and 22 normal colon tissue samples. Since each cell is either a tumor or a normal cell, we removed the outlier logic in both PROCLUS and IPROCLUS algorithms. Figure 2 gives a typical result for the two algorithms. Confusion matrix is defined in the same way as in the PROCLUS paper. Entry (i, j) is equal to the number of data points assigned to output cluster i, which were generated as part of input cluster j. The result is obtained when k is set to 2 since there are only two clusters and l is set to 124, which is based on our experimental analysis. For the Colon Tumor dataset, IPROCLUS can correctly classify 52 out of the 62 tissues, achieving the accuracy of 83.9%, while PROCLUS can only achieve the accuracy of 53.2% (33 correctly classified). We can see that IPROCLUS can achieve much better accuracy on the Colon Tumor dataset than PROCLUS.

The same algorithms are applied to the Leukemia dataset. The Leukemia dataset consists of the expression values for 7129 genes of 47 acute lymphoblastic leukemia (ALL) samples and 25 acute myeloid leukemia (AML) samples. Figure 3 gives a typical result for the two algorithms. IPROCLUS achieves an excellent accuracy rate of 97.2% (70 out of 72 correctly classified). PROCLUS also achieves a higher accuracy rate compared with the Colon Tumor dataset case. However, its accuracy rate, 72.2% (52 out of 72), is still much lower than that of IPROCLUS.

FUTURE TRENDS

Projected clustering in high dimensional space is an interesting research topic and a number of algorithms have been proposed. Biological datasets are always in high dimensional space, which gives projected clustering algorithms a natural advantage in dealing with biological datasets. Several algorithms have been proposed to cluster gene expression data (Yu, Chung, Chan & Yuen, 2004; Bouguessa & Wang, 2007). We expect to see more work on applying projected clustering algorithms on biological datasets. In order to

Figure 3. Confusion matrices on the leukemia dataset

Input / Output	ALL	AML
1	45	2
2	0	25

(a) IPROCLUS

Input / Output	ALL	AML
1	44	3
2	17	8

(b) PROCLUS

make projected clustering algorithms practical, future research needs to remove the dependence on user-chosen parameters and capture biological significant information so that the resulting clusters are more meaningful for biologists. Another direction is called semi-supervised projected clustering (Yip, Cheung, & Ng, 2005; Tang, Xiong, Zhong, & Wu, 2007) which uses limited domain knowledge to aid unsupervised projected clustering process.

CONCLUSION

We have proposed an effective and efficient projected clustering algorithm, IPROCLUS, which is based on PROCLUS. We have significantly improved the accuracy by proposing the modified Manhattan segmental distance. We have reduced the dependence on user input *l* by adding the dimension tuning process at the end of the refinement phase and we have proposed a simplified replacing logic in the iterative phase to offset the running time increase caused by the dimension tuning process.

Empirical results have shown that IPROCLUS is able to accurately discover clusters embedded in lower dimensional subspaces. For the synthetic datasets, it can achieve much higher accuracy and lower dependence on *l* than PROCLUS. We have also applied our algorithm on two biological datasets: the Colon Tumor dataset and the Leukemia dataset. IPROCLUS still achieves much higher accuracy than PROCLUS.

REFERENCES

Aggarwal, C. C., Procopiuc, C., Wolf, J. L., Yu, P. S., & Park, J. S. (1999). Fast algorithms for projected clustering. *ACM SIGMOD International Conference on Management of Data*.

Aggarwal, C. C., & Yu, P. S. (2000). Finding generalized projected clusters in high dimensional spaces. *ACM SIGMOD International Conference on Management of Data*.

Agrawal, R., Gehrke, J., Gunopulos, D., & Raghavan, P. (1998). Automatic subspace clustering of high dimensional data for data mining applications. *ACM SIGMOD International Conference on Management of Data*.

Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D. et al. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences of U.S.A.*, 96, 6745-6750.

Berkhin, P. (2002). Survey of clustering data mining techniques. *Accrue Software*.

Beyer, K., Goldstein, J., Ramakrishnan, R., & Shaft, U. (1999). When is "nearest neighbor" meaningful? *Proceeding of the 7th International Conference on Database Theory*, 217 – 235.

Bouguessa, M., & Wang, S. (2007). PCGEN: A practical approach to projected clustering and its application to gene expression data. *Proceedings of the 2007 IEEE Symposium on Computational Intelligence and Data Mining*.

Deng, P., Wu, W., Huang, Y., & Zhang, Z. (2006). A projected clustering algorithm in high dimensional space. *Proceedings of 15th International Conference on Software Engineering and Data Engineering*.

Easter, M., Kriegel, H. P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. et al. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286, 531-537.

Gonzalez, T. (1985). Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38, 293-306.

Guha, S., Rastogi, R., & Shim, K. (1998). CURE: An efficient clustering algorithm for large databases. *Proceedings of ACM SIGMOD International Conference Management of Data*.

Ibaraki, T., & Katoh, N. (1988). Resource allocation problems: algorithmic approaches. *MIT Press*, Cambridge, Massachusetts.

Kohavi, R. & Sommerfield, D. (1995). Feature subset selection using the wrapper method: overfitting and dynamic search space topology. *Proceedings of the 1st International Conference on Knowledge Discovery and Data Mining*.

Liu, M. L., & Mamoulis, N. (2005). Iterative projected clustering by subspace mining. *IEEE Transactions on Knowledge and Data Engineering*, 17(2), 176-189.

Moise, G., Sander, J., & Ester, M. (2006). P3C: a robust projected clustering algorithm. *Proceedings of the Sixth International Conference on Data Mining*.

Ng, E. K. K., Fu, A. W. C., & Wong, R. C. W. (2005). Projective Clustering by histograms. *IEEE Transaction on Knowledge and Data Engineering*, 17(1), 369-383.

Procopiuc, C. M., Jones, M., Agarwal, P. K., & Murali, T. M. (2002). A monte carlo algorithm for fast projective clustering. *ACM SIGMOD International Conference on Management of Data*.

Raymer, M. L., Punch, W. F., Goodman, E. D., Kuhn, L. A., & Jain, A. K. (2000). Dimensionality reduction using genetic algorithms. *IEEE Transactions on Evolutionary Computation*, 4(2), 164-171.

Tang, W., Xiong, H., Zhong, S., & Wu, J. (2007). Enhancing semi-supervised clustering: a feature projection perspective. *Proceedings of the Thirteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Yip, K. Y., Cheung, D. W., & Ng, M. K. (2003). A highly-usable projected clustering algorithm for gene expression profiles. *3rd ACM SIGMOD Workshop on Data Mining in Bioinformatics*.

Yip, K. Y., Cheung, D. W., & Ng, M. K. (2005). On discovery of extremely low-dimensional clustering using semi-supervised projected clustering. *Proceedings of the 21st International Conference on Data engineering*.

Yu, L. T. H., Chung, F. L., Chan, S. C. F., &Yuen, S. M. C. (2004). Using emerging pattern based projected clustering and gene expression data for cancer detection. *Proceedings of the 2nd Asia-Pacific Bioinformatics Conference*.

KEY TERMS

Clustering: Given a set of data points, find groups of data points so that the points within each group are similar to one another and are dissimilar to the data points belonging to other groups..

Density-Based Clustering: The clustering that defines a cluster as a region that has a higher density of data points than its surrounding regions.

Dimension Reduction: A method concerned with removing the dimensions in the entire data set so that the least amount of information is lost.

Distance-Based Clustering: The clustering that defines a cluster as a partition such that the distance between objects within the same cluster is minimized and the distance between objects from different clusters is maximized.

Feature Selection: A technique to find the closely correlated dimensions for all the data and the clusters in such dimensions.

Medoid: The data point which is the nearest to the center of a cluster.

Projected Clustering: Given a set of data points, find clusters and their relevant dimensions from a dataset such that the similarity is reserved as much as possible.

Proximity-Graph-Based Tools for DNA Clustering

Imad Khoury
School of Computer Science, McGill University, Canada

Godfried Toussaint
School of Computer Science, McGill University, Canada

Antonio Ciampi
Epidemiology & Biostatistics, McGill University, Canada

Isadora Antoniano
IIMAS-UNAM, Ciudad de Mexico, Mexico

Carl Murie
McGill University, Canada & McGill University and Genome Quebec Innovation Centre, Canada

Robert Nadon
McGill University, Canada & McGill University and Genome Quebec Innovation Centre, Canada

INTRODUCTION

Clustering is considered the most important aspect of unsupervised learning in data mining. It deals with finding *structure* in a collection of unlabeled data. One simple way of defining clustering is as follows: the process of organizing data elements into groups, called clusters, whose members are similar to each other in some way. Several algorithms for clustering exist (Gan, Ma, & Wu, 2007); proximity-graph-based ones, which are untraditional from the point of view of statisticians, emanate from the field of computational geometry and are powerful and often elegant (Bhattacharya, Mukherjee, & Toussaint, 2005). A proximity graph is a graph formed from a collection of elements, or points, by connecting with an edge those pairs of points that satisfy a particular neighbor relationship with each other. One key aspect of proximity-graph-based clustering techniques is that they may allow for an easy and clear visualization of data clusters, given their geometric nature. Proximity graphs have been shown to improve typical instance-based learning algorithms such as the *k*-nearest neighbor classifiers in the typical nonparametric approach to classification (Bhattacharya, Mukherjee, & Toussaint, 2005). Furthermore, the most powerful and robust methods for clustering turn out to be those based on proximity graphs (Koren, North, & Volinsky, 2006). Many examples have been shown where proximity-graph-based methods perform very well when traditional methods fail miserably (Zahn, 1971; Choo, Jiamthapthaksin, Chen, Celepcikay, Giusti, & Eick, 2007)

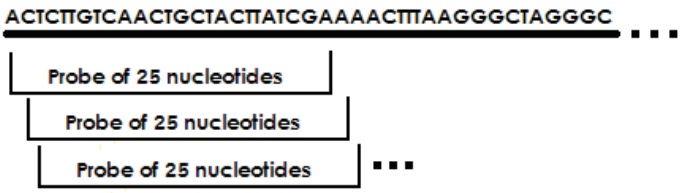
The most well-known proximity graphs are the nearest neighbor graph (*NNG*), the minimum spanning tree (*MST*), the relative neighborhood graph (*RNG*), the Urquhart graph (*UG*), the Gabriel graph (*GG*), and the Delaunay triangulation (*DT*) (Jaromczyk, & Toussaint, 1992). The specific order in which they are introduced is an inclusion order, i.e., the first graph is a subgraph of the second one, the second graph is a subgraph of the third and so on. The *NNG* is formed by joining each point by an edge to its nearest neighbor. The *MST* is formed by finding the minimum-length tree that connects all the points. The *RNG* was initially proposed as a tool for extracting the shape of a planar pattern (Jaromczyk, & Toussaint, 1992), and is formed by connecting an edge between all pairs of distinct points if and only if they are relative neighbors. Two points A and B are relative neighbors if for any other point C, the maximum of d(A, C), d(B, C) is greater than d(A, B), where d denotes the distance measure. A triangulation of a set of points is a planar graph

connecting all the points such that all of its faces, except for the outside face, are triangles. The *DT* is a special kind of triangulation where the triangles are as "fat" as possible, i.e., the circumcircle of any triangle does not contain any other point in its interior. The *UG* is obtained by removing the longest edge from each triangle in the *DT*. Finally, the *GG* is formed by connecting an edge between all pairs of distinct points if and only if they are Gabriel neighbors. Two points are Gabriel neighbors if the hyper-sphere that has them as a diameter is empty, i.e., if it does not contain any other point in its interior. Clustering using proximity graphs consists of first building a proximity graph from the data points. Then, edges that are deemed long are removed, according to a certain edge-cutting criterion. Clusters then correspond to the connected components of the resulting graph. One edge-cutting criterion that preserves Gestalt principles of perception was proposed in the context of *MST*s by C. T. Zahn (Zahn, 1971), and consists in breaking those edges *e* that are at least say, twice as long as the average length of the edges incident to the endpoints of *e*. It has been shown that using the *GG* for clustering, or as part of a clustering algorithm, yields the best performance, and is adaptive to the points, in the sense that no manual tweaking of any particular parameters is required when clustering point sets of different spatial distribution and size (Bhattacharya, Mukherjee, & Toussaint, 2005).

The applications of proximity-graph-based clustering, and of clustering in general, are numerous and varied. Possibilities include applications in the fields of marketing, for identifying groups of customers with similar behaviours; image processing, for identifying groups of pixels with similar colors or that form certain patterns; biology, for the classification of plants or animals given their features; and the World Wide Web, for classifying Web pages and finding groups of similar user access patterns (Dong, & Zhuang, 2004). In bioinformatics, scientists are interested in the problem of DNA microarray analysis (Schena, 2003), where clustering is useful as well. Microarrays are ordered sets of DNA fragments fixed to solid surfaces. Their analysis, using other complementary fragments called probes, allows the study of gene expression. Probes that bind to DNA fragments emit fluorescent light, with an intensity that is positively correlated, in some way, to the concentration of the probes. In this type of analysis, the calibration problem is of crucial importance. Using an experimental data set, in which both concentration and intensity are known for a number of different probes, one seeks to learn, in a supervised way, a simple relationship between intensity and concentration so that in future experiments, in which concentration is unknown, one can infer it from intensity. In an appropriate scale, it is reasonable to assume a linear relationship between intensity and concentration. However, some features of the probes can also be expected to have an effect on the calibration equation; this effect may well be non-linear. Arguably, one may reason that if there is a natural clustering of the probes, it would be desirable to fit a distinct calibration equation for each cluster, in the hope that this would be sufficient to take into account the impact of the probes on calibration. This hope justifies a systematic application of unsupervised learning techniques to features of the probes in order to discover, such a clustering, if it exists.

The main concern remains whether one is able to discover the absence or presence of any real clustering of the probes. Traditionally, clustering of microarray probes has been based on standard statistical approaches, which were used to validate an empirically found clustering structure; however, they were usually complex and depended on specific assumptions (Johnson, & Wichern, 2007). An alternative approach

Figure 1.Probes of 25 nucleotides to be clustered. Shown is a gene sequence and a probe window sliding by one nucleotide.



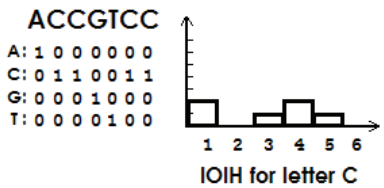
based on proximity graphs could be used, which has the advantage of being relatively simple and of providing a clear visualization of the data, from which one can directly determine whether or not the data support the existence of clusters.

BACKGROUND

A probe is a sequence of a fixed number of nucleotides, say 25 (Fig. 1), and it could simply be regarded as a string of 25 symbols from the alphabet {A, C, G, T}.

Probes are generated by sliding over, by one, a window of size 25 on the same nucleotide sequence. This procedure is not typical but particular to the Mei et. al dataset (Mei, Hubbell, Bekiranov, Mittmann, Christians, Shen, Lu, Fang, Liu, Ryder, Kaplan, Kulp, & Webster, 2003). In order to cluster the probes, one can either resort to classical clustering techniques, or to proximity-graph-based ones, or both. Moreover, for each of these sets of techniques, two approaches can be considered: the feature-based approach and the sequence-based approach. The first one builds the dataset from characteristics extracted from the probes, while the second one deals with the probe sequence directly. In the feature-based approach, classical probe features can be computed such as the frequency of different nucleotide types in the whole probe, and in either of its halves (Johnson, & Wichern, 2007). But current interdisciplinary work on the research theme reveals novel and creative probe-feature-extraction methods. One may, for instance, combine tools from computational music theory and bioinformatics, by considering features inspired from the rhythmic analysis proposed by Gouyon et. al. (Gouyon, Dixon, Pampalk, & Widmer, 2004) and extracting them from microarray probes. In this approach, for each probe, an inter-onset interval histogram (*IOIH*) (Toussaint, 2004; Toussaint, 2005) is built for each letter of the alphabet, and 8 features are computed for each histogram, namely the mean, the geometric mean, the total energy, the centroid, the flatness, the kurtosis, the high-frequency content, and the skewness (Gouyon, Dixon, Pampalk, & Widmer, 2004). The *IOIH* is a histogram that summarizes how many intervals there exist in a binary string of a fixed length, where 'interval' denotes the distance between two not necessarily successive set bits (or '1's). In the left-hand side of Fig. 2, four binary strings are first generated from the DNA sequence, one for each letter, by writing a '1' where that letter occurs and a '0' where it does not. Only the *IOIH* for letter C is shown, in the right-hand side of Fig. 2.

Figure 2. Example of inter-onset interval histogram. We see that there are two inter-onset intervals of length 1, one inter-onset interval of length 3, two inter-onset intervals of length 4, one inter-onset interval of length 5 and no inter-onset intervals of lengths 2, 6.



Then, 6 inter-histogram distances are computed using the Kolmogorov distance. Hence, in total, 32+6 = 38 features are extracted for each probe. Six classical distances are used for defining the dissimilarity of the points in the feature space. They are the standardized and unstandardized Manhattan distance, the standardized and unstandardized Euclidean distance, the Mahalanobis distance and the Mahalanobis-Manhattan distance (Johnson, & Wichern, 2007). The Manhattan distance is the distance between two points measured along axes at right angles. The Mahalanobis distance is effectively a weighted Euclidean distance where the weighting is determined by the sample correlation matrix of the point set.

The second approach is based entirely on the sequence of the symbols in the probe, and aims at producing a distance matrix that summarizes the distances between all pairs of probes, and which serves as input for a clustering algorithm. Sequence-based distances with normalization variations can be derived. These include the nearest neighbour distance, the edit distance (Levenshtein, 1966) and the directed swap distance (Colannino, & Toussaint, 2005). The nearest neighbor distance measures the dissimilarity of two binary strings via the concept of nearest set bit neighbor mapping. A first pass is done over the first string, say from left to right, and set bits are mapped to their closest set-bit neighbors in the second string, in terms of character count. The same kind of mapping is done in a second pass over the second string. The nearest neighbor distance is the accumulated distances

of each mapping link without counting double links twice. The edit distance is a metric that measures the dissimilarity of two strings by counting the minimum number of editing operations needed to transform one string into another. The editing operations typically considered are 'replace', 'insert' and 'delete'. A cost can be associated with each operation, hence penalizing the use of one operation over another. Finally, the directed swap distance measures the dissimilarity of two binary strings via the concept of an assignment. It is equal to the cost of the minimum-cost assignment between the set bits of the first string and the set bits of the second string, where cost is taken to be the minimum number of swaps between adjacent bits needed to displace a set bit in the first string to the position of a set bit in the second string, thereby making one assignment.

Classical clustering techniques include the k-medoids clustering using partitioning around medoids (*PAM*) (Handl, & Knowles, 2005), the hierarchical clustering using single linkage, and classical multidimensional scaling (*CMDS*). *PAM* is an algorithm that clusters around medoids. A medoid is the data point which is the most centrally located in a point set. The sum of the distances of this point to all other points in a point set is less than or equal to the sum of the distances of any other point to all other points in the point set. *PAM* finds the optimal solution. It tends to find 'round' or 'spherical' clusters, and hence is not very efficient when the clusters are in reality elongated, or in line patterns. Hierarchical clustering is a traditional clustering technique that clusters in an agglomerative approach. First, each point is assigned to its own cluster and then, iteratively, the two most similar clusters are joined, until there is only one cluster left. If the 'single linkage' version is used, it becomes equivalent to clustering with minimum spanning trees. Other options include 'complete linkage' and 'average linkage'. Finally, *CMDS* is an algorithm that takes as input a distance matrix and returns a set of points such that the Euclidean distances between them approximate the corresponding values in the distance matrix. This method, in the context of clustering, allows one to try different dimensions in which clustering can be performed. On the other hand, it is a technique to reduce the dimensionality of the data to one that can be easily visualized.

As to proximity-graph-based clustering techniques, they are: *ISOMAP* (Tenenbaum, de Silva, Langford, 2000), and clustering using Gabriel graphs with the Zahn edge-cutting criterion. *ISOMAP* is a proximity-graph based algorithm similar in its goal to *CMDS*, but with the flexibility of being able to learn a broad class of nonlinear manifolds. A manifold is an abstract space in which every point has a neighborhood which resembles the Euclidean space, but in which the global structure may be more complicated. The idea of dimension is important in manifolds. For instance, lines are one-dimensional, and planes two-dimensional. *ISOMAP* is computationally efficient and ensures global optimality and asymptotic convergence. It tries to conserve the geodesic distances between the points, and for that it constructs the *k*-nearest neighbor graph. This set of techniques can help to find and visualize the presence or absence of any real clustering in the data.

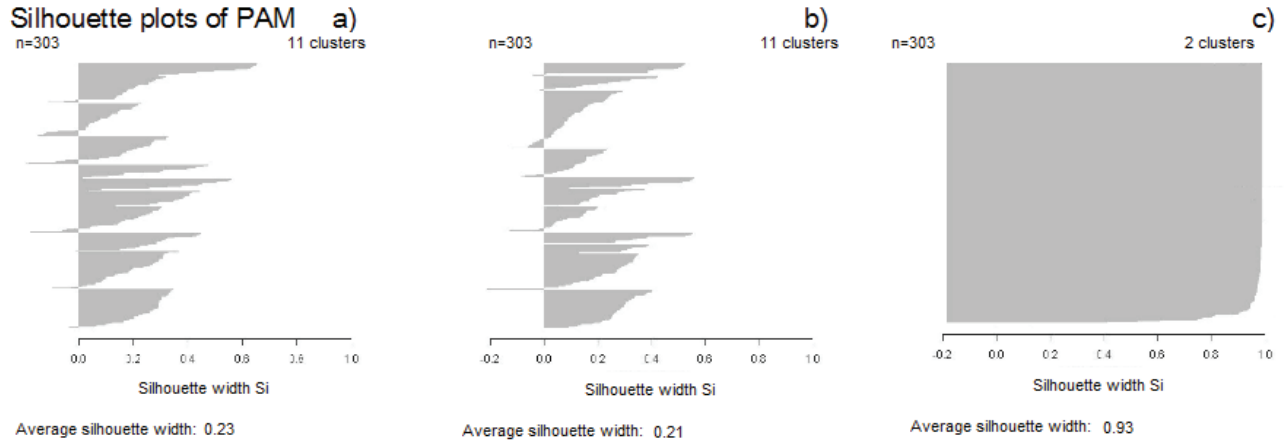
MAIN FOCUS

The rhythmic-analysis-based feature extraction applied to microarray probes, and the nearest neighbor distance for measuring probe dissimilarity are, in themselves, a novelty in the current research on microarray data mining. What is further emphasized here is the comparison between traditional clustering techniques and novel proximity-graph-based clustering techniques, the latter which give a simple and clear way of visualizing the clustering and, for our DNA microarray clustering example, it shows that the data lacks any real clustering.

Dataset

The dataset consists of a collection of genes, with sliding probes (Fig. 1) for each gene. The HTC (Human Test Chip) data set is used. This is a proprietary data set that Affymetrix used in its probe selection algorithm study (Mei, Hubbell, Bekiranov, Mittmann, Christians, Shen, Lu, Fang, Liu, Ryder, Kaplan, Kulp, & Webster, 2003) and was accessed through Simon Cawley and Teresa Webster (Affymetrix). A set of 84 transcripts (77 human and 7 bacterial) were used for probe selection analysis. Each human gene has approximately 500 probes made of 25 nucleotides each. Let us consider one representative gene from which 303 probes have been generated. The dataset is therefore the collection of the 303 probes.

Figure 3. PAM silhouettes using: a) Standardized Manhattan distance, b) Standardized Eucidean distance, c) Mahalanobis distance. A silhouette plot is a plot that displays a measure of how close each point in one cluster is to points in the neighboring clusters. No good clustering of the probes is found, as the average silhouette widths are either small (a and b), or they are high but with only 2 disproportionate clusters found (c).



Methods

PAM and hierarchical clustering using single linkage are used as a baseline to which proximity-graph-based clustering methods are compared.

Clustering of Microarray Probes using Classical Techniques

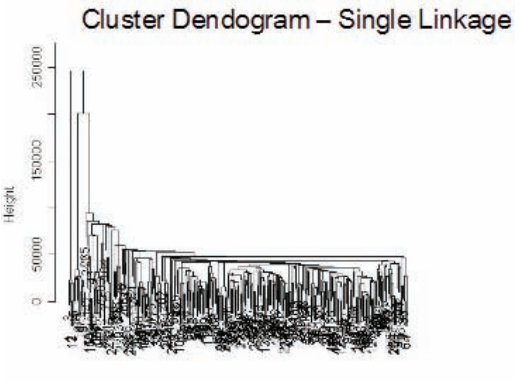
First, *PAM* is applied using the feature-based approach. No good clustering is obtained. Fig. 3 shows a silhouette plot for each of the three distance measures used. A silhouette plot is a plot that displays a measure of how close each point in one cluster is to points in the neighboring clusters. It takes into account both cohesion and separation. A detailed definition is given by (Choo, Jiamthapthaksin, Chen, Celepcikay, Giusti, & Eick, 2007). It is therefore an indicator of the 'goodness' of the clustering. The plots show that clusters do not look well defined; they hence give a first indication that there is no cluster structure in the data.

Next, hierarchical clustering with single linkage, using the feature-based approach, is applied. Fig. 4 shows one representative dendrogram. A dendrogram is a tree diagram used to illustrate the arrangement of the clusters produced by the hierarchical clustering algorithm. The leafs represent the data points, children of the same parent are points in the same cluster, and the edge lengths correspond to the distances between clusters. The dendrogram in Fig. 4 shows, again, that no clustering structure seems to be present in the data, this time with long clusters, just as with spherical ones.

Now, let us apply *PAM* again, but with the sequence-based approach. The total average silhouette width was chosen as an indicator of the 'goodness' of the clustering using *PAM*. With the non-normalized edit distance, a clustering with 61 clusters yielded the best average width. With the nearest neighbor distance, a clustering with 5 clusters yielded the best average width. In both cases, however, the corresponding silhouette plots showed no real clustering. The dendrograms output by the hierarchical clustering with single linkage algorithm also showed no clustering for both the edit distance and the nearest neighbor distance.

The last method we can apply in classical clustering is *CMDS*. As previously defined, *CMDS* can reduce the dimensionality of the data to one that can be visualized. However, clustering would have to be visually performed, which is often difficult and inaccurate. Fig. 5 shows the probe set in a three dimensional space after *CMDS* is applied.

Figure 4. Representative dendrogram for single-linkage clustering, using the unstandardized Manhattan distance. No underlying good clustering is apparent.



Clustering of Microarray Probes using Proximity Graphs

Now, proximity-graph-based clustering techniques are applied on the same dataset and using the same two approaches. In the feature-based approach, clustering using the Gabriel graph – Zahn's criterion gives rise to the plots in Fig. 6. The result, for each distance measure, consists of one very large cluster, containing most of the probes and one or more smaller clusters containing too few probes to be considered. Therefore, we can consider this as more evidence to the theory that no natural clustering structure is present in the data. This time, edges allow an easy visualization of the clustering.

For the sequence-based approach, the *ISOMAP* algorithm is first used to embed the distance matrix in the best Euclidean space, taking into account the possibility that the data set is in reality manifold-shaped. *ISOMAP* found the 3D space to be the best space in which to embed our probes. This makes it possible to plot the points in 3D and to apply clustering using the Gabriel graph – Zahn's criterion algorithm as shown next in Fig. 7. If a higher dimensional space was found by *ISOMAP*, clustering using Gabriel graphs would still be possible, and visualizing it would require additional projection methods. Again, no good clustering is found, as one cluster turns to have most of the points, and the others only few. This time, the absence of real clustering in the data is confirmed.

FUTURE TRENDS

Proximity graphs will continue to play a crucial role in applications such as the calibration problem in microarray analysis, as well as in other clustering applications, especially in those applications where clustering is a preprocessing step for a problem where the need to discover the presence or absence of any real clustering in the data, and where the visualization of the data points and clusters plays a determining role to this end. The current most efficient manifold-learning algorithms are based on nearest neighbor types of graphs. Further research on incorporating Gabriel graphs in manifold learning algorithms, such as *ISOMAP*, should be considered, since Gabriel graphs have consistently been proven to be powerful and helpful tools in designing unsupervised learning and supervised instance-based learning algorithms. Moreover, answers to open questions that arise when

Figure 5. Classical mutidimensional scaling with: a) Edit distance (non-normalized), b) Nearest neighbor distance. No clustering is noticeable.

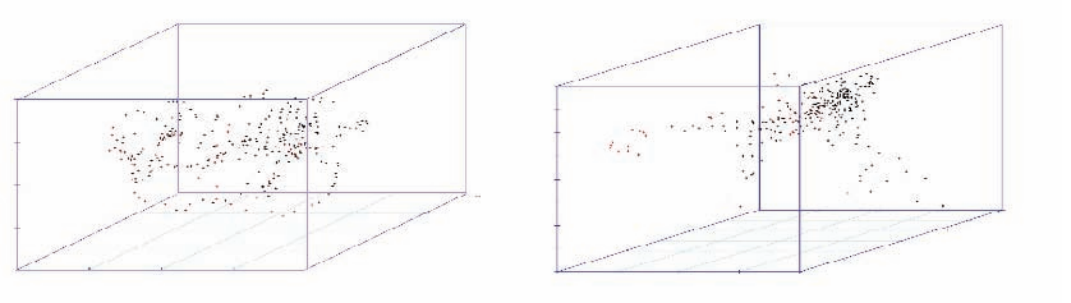


Figure 6.Gabriel graph with Zahn's edge cutting criterion: a) Standardized Manhattan distance, b) Standardized Euclidean distance, c) Mahalanobis distance. No real clustering of the probes is noticeable.

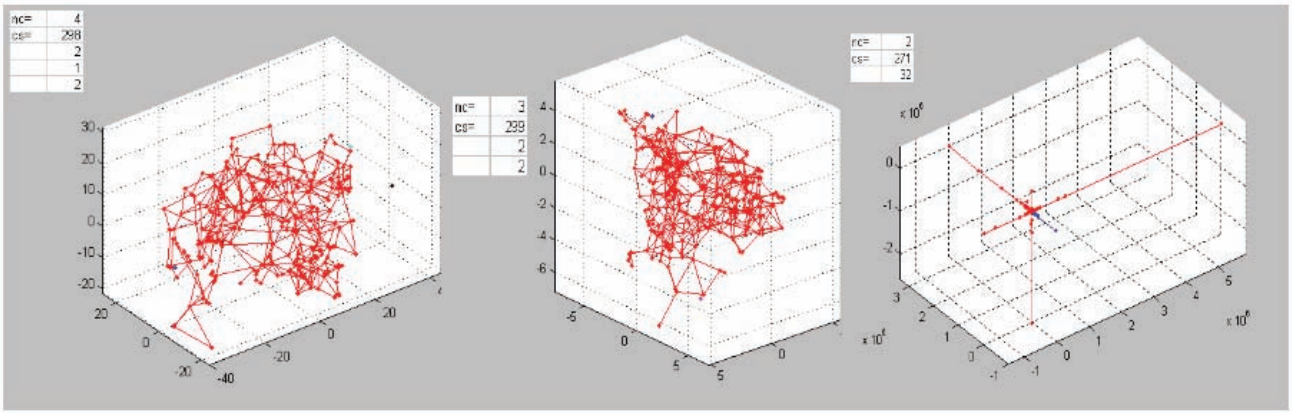
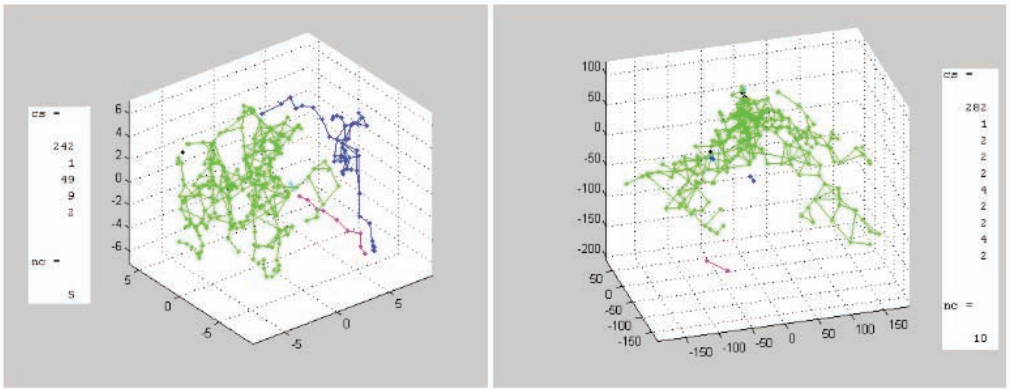


Figure 7. Gabriel Graph with Zahn edge cutting criterion: a) Edit distance, b) Nearest neighbor distance. No real clustering of the probes is noticeable.



applying Gabriel-graph-based clustering on a fixed-size set of points as the dimension gets higher, will have to be investigated. In fact, in a situation like this, Gabriel graphs can have edges between all pairs of points (Chazelle, Edelsbrunner, Guibas, Hershberger, Seidel, & Sharir, 1990; Jaromczyk, & Toussaint, 1992); a situation that may also be viewed from the standpoint of supervised learning, when features of a dataset, which correspond to dimensions, are so numerous, that a comparatively smaller number of data points is not enough to learn them, essentially leading to what is termed the curse of dimensionality. Which subgraph of the Gabriel graph would give the best result in this case, or how sparse the graph should be, will then be interesting future research problems to be solved in this field.

CONCLUSION

Proximity-graph-based clustering can be a very helpful preprocessing tool for the calibration problem in microarray analysis. Both classical and proximity-graph-based clustering methods can be used to cluster microarray probes. However, classical methods do not provide a simple, elegant, and clear way of visualizing the clustering, if it exists. Furthermore, unlike some proximity-graph-based algorithms, they almost always

fail to detect any clusters of structurally complex higher level shapes, such as a manifold. Proximity-graph-based clustering methods can hence be efficient and powerful alternate or complementary tools for traditional unsupervised learning. These methods can also play a useful role in visualizing the absence (or presence) of any real clustering in the data that may have been found using classical clustering methods. Moreover, in this context, novel interdisciplinary probe-feature-extraction methods are being considered, and a sequence-based approach that defines novel distance measures between probes is currently under investigation.

REFERENCES

Bhattacharya, B., Mukherjee, K., & Toussaint, G. T. (2005). Geometric decision rules for high dimensions. *In Proceedings of the 55th Session of the International Statistical Institute.* Sydney, Australia.

Chazelle, B., Edelsbrunner, H., Guibas, L. J., Hershberger, J. E., Seidel, R., & Sharir, M. (1990).

Slimming down by adding; selecting heavily covered points. *Proceedings of the sixth annual symposium on Computational Geometry* (pp. 116-127). Berkley, California, United States.

Choo, J., Jiamthapthaksin, R., Chen, C., Celepcikay, O. U., Giusti, C., & Eick, C. F. (2007). MOSAIC: A Proximity Graph Approach for Agglomerative Clustering. In *Data Warehousing and Knowledge Discovery* of *Lecture Notes in Computer Science* (pp. 231-240). Regensburg, Germany: Springer Berlin / Heidelberg.

Colannino, J., & Toussaint, G. T. (2005). An algorithm for computing the restriction scaffold assignment problem in computational biology. *Information Processing Letters*, *95*(4), 466-471.

Dasarathy, B. V., Sanchez, J. S., & Townsend, S. (2000). Nearest neighbor editing and condensing tools - synergy exploitation. *Pattern Analysis and Applications, 3,* 19-30.

Dong, Y., & Zhuang Y. (2004). Fuzzy hierarchical clustering algorithm facing large databases. *Fifth World Congress on Intelligent Control and Automation: Vol. 5* (pp. 4282 - 4286).

Johnson, R. A., & Wichern, D. W. (Ed.) (2007). *Applied Multivariate Statistical Analysis.* New York, NY: Prentice Hall.

Gan, G, Ma, C., & Wu, J. (2007). Clustering Algorithms. In ASA-SIAM Series on Statistics and Applied Probability . Philadelphia, PA.: SIAM.

Gouyon, F., Dixon, S., Pampalk, E., & Widmer, G. (2004). Evaluating rhythmic descriptors for musical genre classification. *ES 25th International Conference* (pp. 17-19). London, UK.

Handl, J., & Knowles, J. (2005). Multiobjective clustering around medoids. *Proceedings of the Congress on Evolutionary Computation: Vol. 1* (pp. 632-639).

Jaromczyk, J. W., & Toussaint, G. T. (1992). Relative neighborhood graphs and their relatives. *Proceedings of the Institute of Electrical and Electronics Engineers : Vol. 80. No. 9* (pp. 1502-1517).

Koren, Y., North, S. C., & Volinsky, C. (2006). Measuring and extracting proximity in networks. *Proceedings of the 12th ACM SIGKDD international conference on Knowledge Discovery and Data mining :* (pp. 245-255).

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics - Doklady*, *10*(8), 707-710.

Mei, R., Hubbell, E., Bekiranov, S., Mittmann, M., Christians, F. C., Shen, M. M., Lu, G., Fang, J., Liu, W. M., Ryder, T., Kaplan, P., Kulp, D., & Webster, T. A. (2003). Probe selection for high-density oligonucleotide arrays. *Proceedings of the National Academy of Sciences of the United States of America: Vol.* 100. (pp. 11237-11242).

Sanchez, J. S., Pla, F., & Ferri, F. J. (1997). On the use of neighborhood-based non-parametric classifiers. *Pattern Recognition Letters*, *18*, 1179-1186.

Sanchez, J. S., Pla, F., & Ferri, F. J. (1997). Prototype selection for the nearest neighbor rule through proximity graphs. *Pattern Recognition Letters*, *18*, 507-513.

Sanchez, J. S., Pla, F., & Ferri, F. J. (1998). Improving the k-NCN classification rule through heuristic modifications. *Pattern Recognition Letters*, *19*, 1165-1170.

Schena, M. (Ed.). (2003). *Microarray Analysis*. New York, NY: John Wiley & Sons.

Tenenbaum, J. B., de Silva, V., Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science, 290*(5500), 2319-2323.

Toussaint, G. T. (2004). Computational geometric aspects of musical rhythm. *Abstracts of the 14th*

Annual Fall Workshop on Computational Geometry (pp. 47-48). Massachussetts Institute of

Technology.

Toussaint, G. T. (2005). The geometry of musical rhythm. In J. Akiyama et al. (Eds.), *Proceedings of the Japan Conference on Discrete and Computational Geometry: Vol. 3742. Lecture Notes in Computer Science* (pp. 198-212). Berlin, Heidelberg: Springer-Verlag.

Zahn, C. T. (1971). Graph theoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on Computers, C-20*(1), 68-86.

KEY TERMS

Clustering: Data mining technique falling under the unsupervised learning category. It is the process of organizing data elements into groups, called clusters, whose members are similar in some way

Gabriel Graph (GG): A proximity graph formed by joining by an edge all Gabriel neighbors.

Gabriel Neighbors: Two points are Gabriel neighbors if the hyper-sphere that has them as diameter is empty, i.e., if it does not contain any other point.

Inter-Onset Interval Histogram (IOIH): A histogram that summarizes how many intervals there exist in a binary string of a fixed length, where 'interval' denotes the distance between two (not necessarily successive) set bits.

Microarray: An array of ordered sets of DNA fragments fixed to solid surfaces. Their analysis, using other complementary fragments called probes, allows the study of gene expression.

Nearest Neighbor: The point, in a point set, that has the minimum distance to a given point, with respect to a certain distance measure.

Nearest Neighbor Distance: A distance measure that measures the dissimilarity of two binary strings via the concept of nearest set bit neighbor mapping. A first pass is done over the first string, say from left to right, and set bits are mapped to their closest set-bit neighbors in the second string, in terms of character count. The same kind of mapping is done in a second pass over the second string. The nearest neighbor distance is the accumulated distances of each mapping link without counting twice double links.

Nucleotide: A subunit of DNA or RNA. Thousands of nucleotides are joined in a long chain to form a DNA or an RNA molecule,. One of the molecules that make up a nucleotide is a nitrogenous base (A, G, C, or T in DNA; A, G, C, or U in RNA); hence a nucleotide sequence is written as a string of characters from these alphabets.

Probe: A sequence of a fixed number of nucleotides used for the analysis of microarrays. It is designed to bind to specific DNA fragments in a microarray, and emit fluorescent light as an indicator of the binding strength.

Proximity Graph: A graph constructed from a set of geometric points by joining by an edge those points that satisfy a particular neighbor relationship with each other. The most well-known proximity graphs are the nearest neighbor graph (*NNG*), the minimum spanning tree (*MST*), the relative neighborhood graph (*RNG*), the Urquhart graph (*UG*), the Gabriel graph (*GG*), and the Delaunay triangulation (*DT*).

Pseudo-Independent Models and Decision Theoretic Knowledge Discovery

Yang Xiang
University of Guelph, Canada

INTRODUCTION

Graphical models such as Bayesian networks (BNs) (Pearl, 1988; Jensen & Nielsen, 2007) and decomposable Markov networks (DMNs) (Xiang, Wong., & Cercone, 1997) have been widely applied to probabilistic reasoning in intelligent systems. Knowledge representation using such models for a simple problem domain is illustrated in Figure 1: Virus can damage computer files and so can a power glitch. Power glitch also causes a VCR to reset. Links and lack of them convey dependency and independency relations among these variables and the strength of each link is quantified by a probability distribution. The networks are useful for inferring whether the computer has virus after checking files and VCR. This chapter considers how to discover them from data.

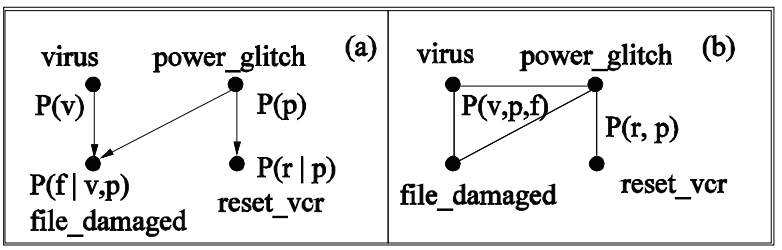
Discovery of graphical models (Neapolitan, 2004) by testing all alternatives is intractable. Hence, heuristic search are commonly applied (Cooper & Herskovits, 1992; Spirtes, Glymour, & Scheines, 1993; Lam & Bacchus, 1994; Heckerman, Geiger, & Chickering, 1995; Friedman, Geiger, & Goldszmidt, 1997; Xiang, Wong, & Cercone, 1997). All heuristics make simplifying assumptions about the unknown data-generating models. These assumptions preclude certain models to gain efficiency. Often assumptions and models they exclude are not explicitly stated. Users of such heuristics may suffer from such exclusion without even knowing. This chapter examines assumptions underlying common heuristics and their consequences to graphical model discovery. A decision theoretic strategy for choosing heuristics is introduced that can take into account a full range of consequences (including efficiency in discovery, efficiency in inference using the discovered model, and cost of inference with an incorrectly discovered model) and resolve the above issue.

BACKGROUND

A graphical model encodes probabilistic knowledge about a problem domain concisely (Pearl, 1988; Jensen & Nielsen, 2007). Figure 1 illustrates a BN in (a) and a DMN in (b). Each node corresponds to a binary variable. The graph encodes dependence assumptions among these variables, e.g., that *f* is directly dependent on *v* and *p*, but is independent of *r* once the value of *p* is observed. Each node in the BN is assigned a conditional probability distribution (CPD) conditioned on its parent nodes, e.g., $P(f \mid v, p)$ to quantify the uncertain dependency. The joint probability distribution (JPD) for the BN is uniquely defined by the product $P(v, p, f, r) = P(f \mid v, p)\, P(r \mid p)\, P(v)\, P(p)$. The DMN has two groups of nodes that are maximally pairwise connected, called *cliques*. Each is assigned a probability distribution, e.g., $\{v, p, f\}$ is assigned $P(v, p, f)$. The JPD for the DMN is $P(v, p, f)\, P(r, p) / P(p)$.

When discovering such models from data, it is important that the dependence and independence relations

Figure 1. (a) An example BN (b) A corresponding DMN



expressed by the graph approximate true relations of the unknown data-generating model. How accurately can a heuristics do so depends on its underlying assumptions.

To analyze assumptions underlying common heuristics, we introduce key concepts for describing dependence relations among domain variables in this section. Let V be a set of discrete variables $\{x_1, \dots, x_n\}$. Each x_i has a finite space $S_{xi} = \{x_{i,j} | 1 \le j \le D_i\}$. When there is no confusion, we write $x_{i,j}$ as x_{ij} . The space of a set $X \subseteq V$ of variables is the Cartesian product $S_X = \prod_{x_i \in X} S_i$. Each element in S_X is a configuration of X, denoted by $x = (x_1, \dots, x_n)$. A probability distribution $P(X)$ specifies the probability $P(x) = P(x_1, \dots, x_n)$ for each x . $P(V)$ is the *JPD* and $P(X)$ ($X \subset V$) is a *marginal distribution.* A probabilistic domain model (PDM) over V defines $P(X)$ for every $X \subseteq V$.

For disjoint subsets W, U and Z of V, W and U are *conditionally independent* given Z, if $P(w \mid u, z) = P(w \mid z)$ for all configurations such that $P(u, z) > 0$. The condition is also denoted $P(W \mid U, Z) = P(W \mid Z)$. It allows modeling of dependency within $W \bigcup U \bigcup Z$ through *overlapping* subsets $W \bigcup Z$ and $U \bigcup Z$.

W and U are *marginally independent* if $P(W \mid U) = P(W)$ holds whenever $P(U) > 0$. The condition allows dependency within $W \bigcup U$ to be modeled over *disjoint* subsets. If each variable x_i in a subset X is marginally independent of $X \setminus \{x_i\}$, then variables in X are *marginally independent*.

Variables in a subset X are *generally dependent* if $P(Y \mid X \setminus Y) \neq P(Y)$ for every $Y \subset X$. For instance, $X = \{x_1, x_2, x_3\}$ is *not* generally dependent if $P(x_1, x_2 | x_3) = P(x_1, x_2)$. It is generally dependent if $P(x_1, x_2 | x_3) \neq P(x_1, x_2)$, $P(x_2, x_3 | x_1) \neq P(x_2, x_3)$ and $P(x_3, x_1 | x_2) \neq P(x_3, x_1)$. Dependency within X cannot be modeled over disjoint subsets but may through overlapping subsets, due to conditional independence in X.

Variables in X are *collectively dependent* if, for each proper subset $Y \subset X$, there exists no proper subset $Z \subset X \setminus Y$ that satisfies $P(Y \mid X \setminus Y) = P(Y \mid Z)$. Collective dependence prevents modeling through overlapping subsets and is illustrated in the next section.

MAIN THRUST OF THE CHAPTER

Pseudo-Independent (PI) Models

A *pseudo-independent* (PI) model is a PDM where proper subsets of a set of collectively dependent variables display marginal independence (Xiang, Wong., & Cercone, 1997). Common heuristics often fail in learning a PI model (Xiang, Wong., & Cercone, 1996). Before analyzing how assumptions underlying common heuristics cause such failure, we introduce PI models below. PI models can be classified into three types: *full, partial, and embedded*. The basic PI model is a full PI model.

Definition 1. A PDM over a set V ($|V| \ge 3$) of variables is a *full PI model* if the following hold:

(S_I) Variables in each proper subset of V are marginally independent.

(S_{II}) Variables in V are collectively dependent.

Example 1 Patient of a chronicle disease changes the health state (denoted by variable s) daily between stable ($s = t$) and unstable ($s = u$). Patient suffers badly in an unstable day unless treated in the morning, at which time no indicator of the state is detectable. However, if treated at the onset of a stable day, the day is spoiled due to side effect. From historical data, patient's states in four consecutive days observe the estimated distribution in Table 1.

The state in each day is uniformly distributed, i.e., $P(s_i = t) = 0.5$ where $1 \le i \le 4$. The state of each day is marginally independent of that of the previous day, i.e., $P(s_i = t | s_{i-1}) = 0.5$ where $2 \le i \le 4$. It is marginally independent of that of the previous two days, i.e., $P(s_i = t | s_{i-1}, s_{i-2}) = 0.5$ where $3 \le i \le 4$. However, states of four days are collectively dependent, e.g., $P(s_4 = u | s_3 = u, s_2 = t, s_1 = t) = 1$, which allows the state of the last day to be predicted from states of previous three days. Hence, the patient's states form a full PI model.

By relaxing condition (S_I), full PI models are generalized into partial PI models defined through *marginally independent partition* (Xiang, Hu, Cercone, & Hamilton, 2000):

Table 1. Estimated distribution of patient health state

(s_1,s_2,s_3,s_4)	P(.)	(s_1,s_2,s_3,s_4)	P(.)
(t, t, t, t)	1/8	(u, t, t, t)	0
(t, t, t, u)	0	(u, t, t, u)	1/8
(t, t, u, t)	0	(u, t, u, t)	1/8
(t, t, u, u)	1/8	(u, t, u, u)	0
(t, u, t, t)	0	(u, u, t, t)	1/8
(t, u, t, u)	1/8	(u, u, t, u)	0
(t, u, u, t)	1/8	(u, u, u, t)	0
(t, u, u, u)	0	(u, u, u, u)	1/8

Definition 2. Let V ($|V| \geq 3$) be a set of variables, and $B = \{ B^1, \ldots, B^m \}$ ($m \geq 2$) be a partition of V. B is a *marginally independent partition* if, for every subset $X = \{x_i^k \mid x_i^k \in B^k, k = 1,\ldots,m\}$, variables in X are marginally independent. Each B^i is a *marginally independent block*.

A marginally independent partition groups variables into m blocks. If a subset X is formed by taking one element from each block, then variables in X are marginally independent. Partial PI models are defined by replacing marginally independent subsets with the marginally independent partition.

Definition 3. A PDM over a set V ($|V| \geq 3$) of variables is a *partial PI model* if the following hold:

(S_I') V can be partitioned into marginally independent blocks.

(S_{II}) Variables in V are collectively dependent.

Table 2 shows the JPD of a partial PI model over $V = \{x_1, x_2, x_3\}$ where x_1 and x_2 are ternary. The marginal probabilities are

$P(x_{1.0}) = 0.3$, $P(x_{1.1}) = 0.2$, $P(x_{1.2}) = 0.5$,
$P(x_{2.0}) = 0.3$, $P(x_{2.1}) = 0.4$, $P(x_{2.2}) = 0.3$,
$P(x_{3.0}) = 0.4$, $P(x_{3.1}) = 0.6$.

The marginally independent partition is $\{\{x_1\}, \{x_2, x_3\}\}$. Variable x_1 is marginally independent of each variable in the other block, e.g., $P(x_1, x_{2.0}) = P(x_{1.1})\,P(x_{2.0}) = 0.06$. However, variables in block $\{x_2, x_3\}$ are dependent, e.g., $P(x_{2.0}, x_{3.1}) = 0.1 \neq P(x_{2.0})\,P(x_{3.1}) = 0.18$. The three variables are collectively dependent, e.g., $P(x_{1.1} \mid x_{2.0}, x_{3.1}) = 0.1$ and $P(x_{1.1} \mid x_{2.0}) = P(x_{1.1} \mid x_{3.1}) = 0.2$.

A partial PI model may involve only a proper subset of V and remaining variables show normal dependency. The subset is an *embedded* PI submodel. A PDM can embed multiple submodels.

Definition 4. Let a PDM be over a set V of generally dependent variables. A proper subset $V' \subset V$ ($|V'| \geq 3$) of variables forms an *embedded* PI submodel if the following hold:

(S_{III}) V' forms a partial PI model.

(S_{IV}) The marginal independent partition $B = \{ B^1, \ldots, B^m \}$ of V' extends into V. That is, V partitions into $\{X^1,\ldots,X^m\}$ such that $B^i \subseteq X^i, (i = 1,\ldots,m)$ and, for each $x \in X_i$ and $y \in X_j$ $(i \neq j)$, x and y are marginally independent.

Table 3 shows the JPD of a PDM with an embedded PI submodel over x_1, x_2 and x_3. The marginal probabilities are $P(x_{1.0}) = 0.3$, $P(x_{2.0}) = 0.6$, $P(x_{3.0}) = 0.3$, $P(x_{4.0}) = 0.34$, $P(x_{5.0}) = 0.59$.

The marginally independent partition of the submodel is $\{B^1 = \{x_1\}, B^2 = \{x_2, x_3\}\}$.

Table 2. A partial PI model where v = (x_1, x_2, x_3)

v	P(.)	v	P(.)	v	P(.)	v	P(.)	v	P(.)	v	P(.)
(0,0,0)	0.05	(0,1,1)	0.11	(1,0,0)	0.05	(1,1,1)	0.08	(2,0,0)	0.10	(2,1,1)	0.11
(0,0,1)	0.04	(0,2,0)	0.06	(1,0,1)	0.01	(1,2,0)	0.03	(2,0,1)	0.05	(2,2,0)	0.01
(0,1,0)	0.01	(0,2,1)	0.03	(1,1,0)	0	(1,2,1)	0.03	(2,1,0)	0.09	(2,2,1)	0.14

Table 3. A PDM with an embedded PI submodel where v =$\{x_1, x_2, x_3, x_4, x_5\}$

v	P(.)	v	P(.)	v	P(.)	v	P(.)
(0,0,0,0,0)	0	(0,1,0,0,0)	.0018	(1,0,0,0,0)	.0080	(1,1,0,0,0)	.0004
(0,0,0,0,1)	0	(0,1,0,0,1)	.0162	(1,0,0,0,1)	.0720	(1,1,0,0,1)	.0036
(0,0,0,1,0)	0	(0,1,0,1,0)	.0072	(1,0,0,1,0)	.0120	(1,1,0,1,0)	.0006
(0,0,0,1,1)	0	(0,1,0,1,1)	.0648	(1,0,0,1,1)	.1080	(1,1,0,1,1)	.0054
(0,0,1,0,0)	.0288	(0,1,1,0,0)	.0048	(1,0,1,0,0)	.0704	(1,1,1,0,0)	.0864
(0,0,1,0,1)	.0072	(0,1,1,0,1)	.0012	(1,0,1,0,1)	.0176	(1,1,1,0,1)	.0216
(0,0,1,1,0)	.1152	(0,1,1,1,0)	.0192	(1,0,1,1,0)	.1056	(1,1,1,1,0)	.1296
(0,0,1,1,0)	.0288	(0,1,1,1,1)	.0048	(1,0,1,1,1)	.0264	(1,1,1,1,1)	.0324

Outside the submodel, B^1 extends to include x_4 and B^2 extends to include x_5. Each variable in one block is marginally independent of each variable in the other block, e.g.,

$P(x_{1,1}, x_{5,0}) = P(x_{1,1})\, P(x_{5,0}) = 0.413.$

Variables in the same block are pairwise dependent, e.g.,

$P(x_{2,1}, x_{3,0}) = 0.1 \neq P(x_{2,1})\, P(x_{3,0}) = 0.12.$

Variables in the submodel are collectively dependent, e.g.,

$P(x_{1,1} \mid x_{2,0}, x_{3,1}) = 0.55,\ P(x_{1,1} \mid x_{2,0}) = P(x_{1,1} \mid x_{3,1}) = 0.7.$

However, x_5 is independent of other variables given x_3, displaying conditional independence, e.g.,

$P(x_{5,1} \mid x_{2,0}, x_{3,0}, x_{4,0}) = P(x_{5,1} \mid x_{3,0}) = 0.9.$

PDMs with embedded PI submodels are the most general PI models.

Heuristics for Model Discovery

Given a data set over *n* variables, the number of possible network structures is super-exponential (Cooper & Herskovits, 1992). To make discovery tractable, a number of heuristics are commonly applied. The most common is the *Naive Bayes* heuristic (Zhang, 2004). It restricts potential models to Naive Bayes models whose graph consists of a single root (the *hypothesis*) and its observable child nodes (the *attributes*). Since the hypothesis is given, discovery focuses on finding the CPD at each node and is very efficient.

Another heuristic is the *TAN* heuristic, that restricts potential models to *tree augmented Naive Bayes* models (Friedman, Geiger, & Goldszmidt, 1997). Its graph also has a single root (the hypothesis). However, attributes themselves form a tree (see Figure 2). Each attribute has the hypothesis and at most one other attribute as its parent nodes.

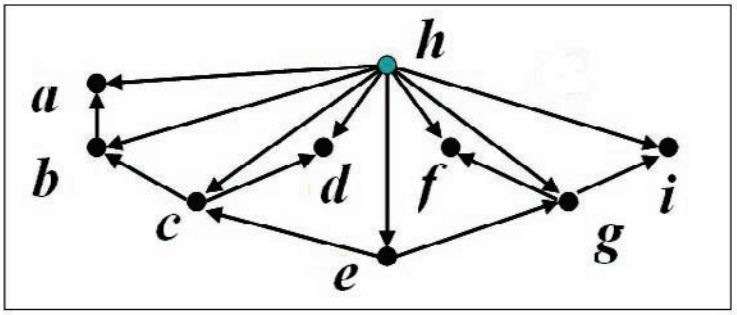
The above heuristics limit the model space. Heuristics below limit the search procedure. One common heuristic is the *single-link lookahead* (Cooper & Herskovits, 1992; Heckerman, Geiger & Chickering, 1995; Lam & Bacchus, 1994). Learning starts with an initial graph. Successive graphical structures, representing different sets of independence assumptions, are adopted. Each adopted structure differs from its predecessor by a single link and improves a score metric optimally.

An alternative is the *bounded multi-link lookahead* (Hu & Xiang, 1997) where an adopted structure differs from its predecessor by up to $k > 1$ links. The algorithm applies single-link lookahead and low-order (small *k*) multi-link lookahead as much as possible, and uses high-order (large *k*) multi-link lookahead only when necessary.

Underlying Assumptions and their Implications

Knowledge discovery starts with a dataset generated by an unknown PDM *M*. The goal is to uncover a graphical model that approximates *M*. The best outcome is often known as the *minimal I-map* (Pearl, 1988). It is a graph *G* whose nodes correspond to variables in

Figure 2. Graph structure of a TAN model where h is the hypothesis



M and whose links are as fewer as possible such that graphical separation among nodes in *G* implies conditional independence in *M*. The assumption underlying a heuristic determines its ability to discover minimal I-maps for various PDMs. The following are assumptions underlying Naïve Bayes and TAN.

Proposition 1. In a Naïve Bayes model, every two attributes are conditionally independent given the hypothesis.

Proposition 2. In a TAN model, every two non-adjacent attributes are conditionally independent given the parent of one of them and the hypothesis.

The general assumption underlying the single-link lookahead is unclear. Known results are based on particular algorithms using the heuristic and are centered around *faithfulness*. A PDM *M* is *faithful* if there exists some graph *G* such that conditional independence among variables in *M* implies graphical separation among corresponding nodes in *G*, and vice versa. Spirtes, Glymour and Scheines (1993) present a sufficient condition: If *M* is faithful, the algorithm in question can discover a minimal I-map of *M*. Xiang, Wong and Cercone (1996) present a necessary condition: If *M* is unfaithful, the output of the algorithm in question will not be an I-map. Hence, faithfulness will be regarded as the primary assumption underlying the single-link lookahead heuristic.

The bounded multi-link lookahead heuristic does not make any of the above assumptions and is the most general among heuristics mentioned above. Implications of these assumptions to discovery of PI models are summarized below (Xiang, 2007).

Theorem 1. Let Λ be the set of all Naive Bayes models and Λ' be the set of all PI models over *V*. Then $\Lambda \cap \Lambda' = \varnothing$.

Theorem 2. Let Λ be the set of all TAN models over *V*. Let Λ' be the set of all PI models over *V* such that each PI model in Λ' contains at least one embedded PI submodel over 4 or more variables. Then $\Lambda \cap \Lambda' = \varnothing$.

Theorem 3. A PI model is unfaithful.

Theorems 1 and 3 say that Naive Bayes and single-link lookahead heuristics cannot discover a minimal I-map if the unknown PDM is PI. Theorem 2 says that if the unknown PDM is beyond the simplest PI model (with exactly 3 variables), then the TAN heuristic cannot discover a minimal I-map.

Suppose that these three heuristics (coupled with known algorithms) are applied to the data in Example 1 in order to find the best strategy for patient treatment. They will be mislead by the marginal independence and return an empty graph (four nodes without links). This is equivalent to say that there is no way that the patient can be helped (either untreated and possibly suffering from the disease, or treated and possibly suffering from the side effect).

On the other hand, if a bounded 6-link lookahead heuristic is applied, the correct minimal I-map will be returned. This is due to the ability of multi-link lookahead to identify collective dependence. Although in this small example, the minimal I-map is a complete graph, the bounded 6-link lookahead can still discover a minimal I-map when the PI model is embedded in a much large PDM. From this discovered model, a targeted treatment strategy can be developed by predict-

ing the patient's state from states of the last three days. Discovery of a PI model from social survey data and experimental result on its performance can be found in (Xiang, Hu, Cercone, & Hamilton, 2000).

FUTURE TRENDS

Decision Theoretic Strategy

Heuristics such as Naïve Bayes, TAN and single-link lookahead are attractive to data mining practitioners due to mostly two reasons: First, they are more efficient. Second, PDMs that violate their underlying assumptions are less likely. For instance, unfaithful PDMs are considered much less likely than faithful ones (Spirtes, Glymour and Scheines, 1993). Although efficiency in discovery and prior probability of potential model are important factors, an additional factor, the cost of suboptimal decision (such as that according to the discovered empty graph for Example 1) has not been paid sufficient attention. A decision theoretic strategy (Xiang, 2007) that integrates all these factors is outlined below, where faithfulness is used as an example assumption.

Let A and A' be alternative discovery algorithms, where A assumes faithfulness and A' does not. Costs of discovery computation are $C_{disc}(A) = d$ and $C_{disc}(A') = d'$, where $d < d'$. The unknown PDM M has a small probability ε to be unfaithful. Choosing A, if M is faithful, the discovered model supports optimal actions. If M is unfaithful, the discovered model causes suboptimal actions. Choosing A', no matter M is faithful or not, the discovered model supports optimal actions. Let the action cost of a correct model (a minimal I-map) be $C_{opt} = 0$ and that of an incorrect model be $C_{sub} = \omega > 0$. The expected cost of choosing A is

$$C_{disc}(A) + (1-\varepsilon) C_{opt} + \varepsilon C_{sub} = d + \varepsilon \omega$$

and that of choosing A' is $C_{disc}(A') + C_{opt} = d'$. According to decision theory, A' is a better choice if and only if

$$\omega > (d' - d)/\varepsilon .$$

In other words, for mission critical applications, where the above inequation often holds, the less efficient but more open-minded algorithm A' should be preferred.

CONCLUSION

Heuristics must be used to render discovery of graphical models computationally tractable. They gain efficiency through underlying assumptions. Naive Bayes makes the strongest assumption, followed by TAN, followed by single-link lookahead, followed by bounded multi-link lookahead, while their complexities are reversely ordered. These assumptions are not subject to verification in the discovery process. The stronger the assumption made, the more likely that the discovered model is not the minimal I-map and, as a result, the model does not support the optimal decision. A decision-theoretic strategy chooses heuristic based on discovery efficiency, likelihood of discovering an incorrect model, as well as consequence in applying an incorrectly discovered model in decision making. For mission critical applications, a more open-minded heuristic should be preferred even though the computational cost of discovery may be higher.

REFERENCES

Cooper, G.F., & Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine Learning, 9*, 309-347.

Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian networks classifiers. *Machine Learning, 29*, 131-163.

Heckerman, D., Geiger, D., & Chickering, D.M. (1995). Learning Bayesian networks: the combination of knowledge and statistical data. *Machine Learning, 20*, 197-243.

Hu, J., & Xiang, Y. (1997). Learning belief networks in domains with recursively embedded pseudo independent submodels, In *Proc. 13th Conf. on Uncertainty in Artificial Intelligence*, (pp. 258-265).

Jensen, F.V., & Nielsen, T.D. (2007). *Bayesian networks and decision graphs* (2nd Ed.). Springer.

Lam, W., & Bacchus, F. (1994). Learning Bayesian networks: an approach based on the MDL principle. Computational Intelligence, 10(3):269--293.

Neapolitan, R.E. (2004). *Learning Bayesian networks*. Prentice Hall.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. Morgan Kaufmann.

Spirtes, P., Glymour, C., & Scheines, R. (1993). *Causation, prediction, and search*. Springer-Verlag.

Wong, S.K.M., & Xiang, Y. (1994). Construction of a Markov network from data for probabilistic inference. *Proc. 3rd Inter. Workshop on Rough Sets and Soft Computing*, 562-569.

Xiang, Y. (2007). A Decision theoretic view on choosing heuristics for discovery of graphical models. In *Proc. 20th Inter. Florida Artificial Intelligence Research Society Conf*., (pp. 170-175).

Xiang, Y., Hu, J., Cercone, N., & Hamilton, H. (2000). Learning pseudo-independent models: Analytical and experimental results. In H. Hamilton, (Ed.), *Advances in Artificial Intelligence*, (pp. 227-239).

Xiang, Y., Lee, J., & Cercone, N. (2003). Parameterization of pseudo-independent models. In *Proc. 16th Inter. Florida Artificial Intelligence Research Society Conf.*, (pp. 521-525).

Xiang, Y., Wong, S.K.M., & Cercone, N. (1996). Critical remarks on single link search in learning belief networks. In *Proc. 12th Conf. on Uncertainty in Artificial Intelligence*, (pp. 564-571).

Xiang, Y., Wong, S.K.M., & Cercone. N. (1997). A `microscopic' study of minimum entropy search in learning decomposable Markov networks. *Machine Learning, 26*(1), 65-92.

Zhang, H. (2004). The optimality of naive Bayes. In *Proc. of 17th International FLAIRS conference (FLAIRS 2004)* (pp. 562-567).

KEY TERMS

Bounded Multi-Link Lookahead Heuristic: It differs from the single-link lookahead in that each adopted graph is selected from candidates that differ from its successor by up to $k>1$ links. It requires higher but bounded computational cost, makes the weakest assumption, and can discover PDMs that are not discoverable by the single-link lookahead such as PI models.

Embedded PI Submodel: An embedded PI submodel is a full or partial PI model over a proper subset of domain variables. The most general PI models are those that embed PI submodels in large problem domains.

Full PI Model: A full PI model is a PDM where every proper subset of variables is marginally independent but the entire set is collectively dependent. They are the most basic PI models.

Naïve Bayes Heuristic: It assumes that the model graph consists of a single root (the *hypothesis*) and its observable child nodes (the *attributes*). It makes the strongest independence assumption and is the most efficient.

Partial PI Model: A partial PI model is otherwise the same as a full PI model except that some subsets of variables may not be marginally independent. A full PI model is also a partial PI model. Hence, partial PI models are more general.

Single-Link Lookahead Heuristic: The discovery process using this heuristic consists of a sequence of adopted graphs such that each is selected from candidates that differ from its successor by exactly one link. Models discoverable with this heuristic are usually faithful PDMs.

TAN Heuristic: It assumes the same as Naïve Bayes plus that each attribute may have at most one other attribute as the additional parent.

Quality of Association Rules by Chi-Squared Test

Wen-Chi Hou
Southern Illinois University, USA

Maryann Dorn
Southern Illinois University, USA

INTRODUCTION

Mining market basket data (Agrawal et al. 1993, Agrawal et al. 1994) has received a great deal of attention in the recent past, partly due to its utility and partly due to the research challenges it presents. Market basket data typically consists of store items purchased on a per-transaction basis, but it may also consist of items bought by a customer over a period of time. The goal is to discover buying patterns, such as two or more items that are often bought together. Such finding could aid in marketing promotions and customer relationship management. Association rules reflect a fundamental class of patterns that exist in the data. Consequently, mining association rules in market basket data has become one of the most important problems in data mining.

Agrawal et al. (Agrawal, et al. 1993, Agrawal et al. 1994) have provided the initial foundation for this research problem. Since then, there has been considerable amount of work (Bayardo et al. 1999, Bayardo et al. 1999, Brin et al. 1997, Han et al. 2000, Park et al. 1995, Srikant et al. 1995, Srikant et al. 1997, Zaki et al. 1997, etc.) in developing faster algorithms to find association rules. While these algorithms may be different in their efficiency, they all use minsup (minimum support) and minconf (minimum confidence) as the criteria to determine the validity of the rules due to their simplicity and natural appeals. Few researchers (Brin et al. 1997, Aumann et al. 1999, Elder, 1999, Tan et al. 2002) have suspected the sufficiency of these criteria. On the other hand, Chi-squared test has been used widely in statistics related fields for independence test. In this research, we shall examine the rules derived based on the support-confidence framework (Agrawal et al. 1993, Agrawal et al. 1994) statistically by conducting Chi-squared tests. Our experimental results show that a surprising 30% of the rules fulfilling the minsup and minconf criteria are indeed insignificant statistically.

BACKGROUND

The task of mining association rules is first to find all itemsets that are above a given minimum support ratio (minsup). Such itemsets are called large or frequent itemsets. Then, association rules are derived based on these frequent itemsets. For example, both {A, B, C, D} and {A, B} are frequent itemsets. The association rule, AB=>CD, is derived if at least c% of the transactions that contain AB also contain CD, where c% is a pre-specified constant called minimum confidence (minconf).

Support-Confidence Framework

We use the example in (Brin et al. 1997) to illustrate the support-confidence framework (Agrawal, et al. 1993, Agrawal et al. 1994). Suppose there are totally 100 transactions. 25 transactions buy tea and among them, 20 transactions also buy coffee. Based on the support-confidence framework, the rule 'tea=> coffee' has a support of 20% (20 / 100) and a confidence of 80% (20 / 25). Suppose minsup =5% and minconf =60%. Then, the rule is validated by the framework.

Chi-Squared Test for Independence

Chi-squared (χ^2) test is a non-parametric statistical method that can be used to test independence among attributes. Compared to the support-confidence framework, it uses more information, such as the numbers of transactions that buy tea but not coffee, buy coffee but not tea, and buy neither coffee nor tea, to determine the independence of attributes.

Table 1. 2 by 2 contingency table

	coffee	no_coffee	∑ row rorow
tea	20	5	25
no_tea	70	5	75
∑ col	90	10	100

In Table 1, we show a 2 by 2 contingency table (Glass, 1984, Gokhale, 1978), which contains more information for the independence test. The chi-square test result indicates that tea and coffee are independent variables since $\chi^2 = 3.703703$ (with degree of freedom = 1) is non-significant at 95% confidence level. In other words, tea is not a determining factor whether people will buy coffee or not, contradicting the rule derived by the support-confidence framework.

Chi-squared test is reliable under a fairly permissive set of assumptions. As a rule of thumb, Chi-squared test is recommend (Glass, 1984, Mason, et al. 1998) only if (1) all cells in the contingency table have expected value greater than 1, and (2) at least 80% of the cells in the contingency table have expected value greater than 5. For the large tables (more than four cells), the usual alternative is to combine or collapse cells (Glass, 1984, Han, et al. 2000, Mason, et al. 1998) when the cells have low values. The potential advantages of the χ^2 statistics (Brin, et al. 1997) over the commonly used support-confidence framework are:

1. The use of the Chi-squared significance test for independence is more solidly grounded in statistical theory. In particular, there is no need to choose ad-hoc values for support and confidence.
2. The Chi-squared statistic simultaneously and uniformly takes into account all possible combinations of the presence and absence of the various attributes being examined as a group.
3. Chi-squared test at a given significance level is upward closed. In other words, if an i-itemset is correlated, all its supersets are also correlated.

MAIN FOCUS

Experimental Design

Four synthetic data sets are generated using the IBM/ Quest data generator (Bayardo et al. 1999), which has been widely used for evaluating association rule mining algorithms. The data sets generated are then fed into CBA/DBII data mining system (Liu et al. 1999) to generate association rules. Finally, Chi-squares tests are conducted on the association rules generated.

The synthetic transactions are to mimic the transactions generated in the retailing environment. A realistic model will address the observation that people tend to buy a number of items together. Thus, transaction sizes are typically clustered around a mean and a few transactions have many items.

In order to model the phenomenon that frequent itemsets often have common items, some fraction of items in subsequent itemsets are chosen from the previous itemset generated. It uses an exponentially distributed random variable (Bayardo et al. 1999) with the mean equal to a given correlation level to decide this fraction for each itemset. The remaining items are picked at random. In this study, the correlation level was set to 0.25. Bayardo et al. (Bayardo et al. 1999) ran some experiments with correlation levels set to 0.5 and 0.75 but did not find much difference in the results.

Each itemset in the database has a weight associated with it, which corresponds to the probability that this itemset will be picked. This weight is picked from an exponential distribution with unit mean, and is then normalized so that the sum of the weights for all the itemsets is 1.

Four synthetic datasets were generated with slightly different parameters, such as the number of transactions, number of items, and average confidence for rules (see Table 2). The other common factors are 'average transaction length (5),' 'number of patterns (100),' 'average length of pattern (4),' and 'correlation between consecutive patterns (0.25).'

Once the data sets are generated, CBA system (Liu et al. 1999, Liu et al. 1999) is used to generate association rules that satisfy the given minsup and minconf.

Table 2. Four datasets

Data Set	Num of Transactions	Num of Items	Avg. Conf. Level
1	2217	20	0.5
2	5188	20	0.5
3	5029	40	0.5
4	5096	40	0.75

Experimental Results

Effects of Minsup and Minconf

The value of minsup controlls the number of frequent itemsets generated. Table 3 shows the reduction of the numbers of frequent itemsets in relation to the increase in the minsup value. For example, for dataset 1, the number of frequent itemsets drops from 3401 to 408 when minsup increases from 1% to 5%. In addition, the size of the dataset could also affect the selection of minsup significantly. The larger the data size, the larger the variety of items, and thus the smaller the support for each itemset. For example, the datasets with over 5,000 transactions, i.e., datasets 2, 3, and 4, generated much smaller numbers of frequent itemsets than dataset 1 with 2,217 transactions (see Table 3) as there is a larger variety of items in the transactions.

The relationship among minsup, minconf, and the number of association rules generated is shown in Table 4 for each dataset. The confidence level affects the number of association rules generated. The higher the level of minconf, the less the numbers of frequent itemsets and association rules sustain. When the data size is large, the lower minsup was preferred as there are too many different items. If the minsup is set too high, combined with high minconfs, such as 75% or 90%, it can often generate zero association rule in the

Table 3. Numbers of frequent itemsets at various minsup levels

Dataset / Minsup	1%	2%	5%	10%	15%	20%
1	3401	1488	408	131	65	36
2	1836	808	246	84	37	24
3	1513	521	118	40	17	11
4	1600	531	118	40	17	11

Table 4. The relationship among minsup, minconf, and the number of the association rules

Dataset	Minsup	1%			2%			5%			10%		
	minconf	50%	75%	90%	50%	75%	90%	50%	75%	90%	50%	75%	90%
1	#Rules	3169	1261	89	1365	353	2	358	38	0	112	3	0
2	#Rules	1487	217	9	626	43	0	176	5	0	57	0	0
3	#Rules	576	247	30	130	16	1	21	0	0	6	0	0
4	#Rules	677	204	64	150	18	3	24	0	0	8	0	0

experiments. For example, for the dataset with 5,188 transactions, i.e., dataset 2 in Table 4, at minsup=1% and minconf=50%, it generated 1487 rules; but when the minsup becomes 2% and minconf=75%, it generated only 43 rules.

Chi-Square Tests for Independence Results

The results of Chi-squared tests constituted the major findings of this study. Note that all Chi-squared tests conducted in this study adopt the commonly used significance level, $p \leq 0.05$, as a cutoff point. The critical value is 3.841 for 1 degree of freedom.

In the following, we use the dataset with 2,217 transactions as an example. Each transaction had three attributes: Transaction number, Customer ID, and Item. There are 20 different items in the transactions.

The first experiment was conducted at minsup=10% and minconf=50%. CBA produced 112 association rules as shown in Table 4. The 112 rules were then tallied into 2 by 2 contingency tables and tested by the Chi-squared for independence. The results showed that 37 out of 112 rules were not significant. In other words, in 37 rules, there is no relationship between the antecedent and the consequent.

With the same dataset, the minsup was reduced to 5% and the minconf kept at the same level of 50%, that is, the restriction was loosened, CBA program produced 385 association rules as shown in Table 4. Out of these 385 rules, 134 rules were not significant. The previous 37 rules were part of these 134 rules. It again showed that no relationship between antecedent and consequent in one third of the association rules. The results demonstrated a striking discrepancy between the results generated by the support-confidence framework and those by the Chi-squared tests.

Chi-squared Test vs. Support-Confidence

We present more detailed comparisons between the results of the support-confidence framework and Chi-squared tests. In the Tables 5, the cell values are the number of transactions. L and R are the itemsets on the left- and right-hand sides of the rule, respectively; NO-L and NO-R are itemsets that do not have any items in L and R, respectively; T_ROW and T_COL are the total numbers of transactions on each row and column, respectively; SUP_L and SUP_R are the supports for the itemsets on the left- and right-hand sides of the rule; CONF is confidence.

With minsup=50% and minconf=1%, Rules 175 and 169 are validated by the support-confidence framework with the same confidence of 60% and support of 5.01% as shown in Tables 5 and 6. However, the Chi-squared tests give different results for these two rules. That is, there is no relationship between L and R in rule 175, but there exists a relation in rule 169. For rule 175, the χ^2 test result was not significant ($p > 0.05$), indicating that 'buying apple and ketchup' is independent from 'buying pamper". For rule 169, the χ^2 result was significant ($p < 0.05$), which indicated that there is a relationship between 'buying an oven' and 'buying apple and ketchup'.

Example 1: Uni-Directional vs. Bi-Directional

The association rule in minsup/minconf is uni-directional. A rule X -> Y does not necessarily imply the rule Y -> X. In addition, it cannot detect negative implication, such as buying product X and not buying product Y. However, the Chi-squared statistic simultaneously and

Table 5. Non-significant relationship

Rule 169: AP/KP -> OV					
	R	NO_R	T_ROW	SUP_L	CONF
L	111	74	185	8.34%	60.00%
NO_L	1037	995	2032		
T_COL	1148	1069	2217		
SUP_R	51.78%			5.01%	
$p \leq 0.019456897$					

Table 6. Significant relationship

Rule 175: AP/KP ->PA					
	R	NO_R	T_ROW	SUP_L	CONF
L	111	74	185	8.34%	60.00%
NO_L	1134	898	2032		
T_COL	1245	972	2217		
SUP_R	56.16%			5.01%	
$p \leq 0.271182286$					

uniformly takes into account all possible combinations of the presence and absence of the various attributes being examined as a group. We shall illustrate these in the following examples.

Example 2

At minconf =50%, Rule 1 is accepted as an association rule while Rule 2 is pruned off because of its low confidence (24.32%). However, the χ^2 test shows both P values were less than 0.05, which it implies that T and S are correlated and S->T should have also be accepted as a rule.

Chi-square test can be used to detect negative implication, illustrated by the following example.

Example 3

As shown in Table 8, the χ^2 values for both L -> R and L -> (NO_R) are the same. Since $p > 0.05$, it indicated that L and R were two independent attributes. However, at minconf =50%, Rule 3 was accepted as an association rule while Rule 4 was pruned off because of low confidence (47.27%).

FUTURE TRENDS

While the support-confidence framework lays the foundation of modern market basket data mining, it has some obvious weakness. We believe incorporating statistical methods into this framework would make it more robust and reliable.

CONCLUSION

This empirical study aimed at evaluating the validity of association rules derived based on the minsup and minconf criteria. The results of this study strongly suggest that minsup and minconf do not provide adequate discovery of useful associations. The results of Chi-squared tests showed

1. Minsup and minconf can significantly cut down the number of rules generated but does not necessarily cut off those insignificant rules.
2. Approximately, one-third of association rules, discovered at minconf = 50% and minsup= 5% or 10%, failed to prove that there is relationship

Table 7. Chi-square Test on bi-directional relationship

Rule 1: T -> S					
	S	NO_S	T_ROW	SUP_L	CONF
T	36	19	55	29.10%	65.45%
NO_T	112	22	134		
T_COL	148	41	189		
SUP_R	78.31%			19.05%	
$\chi^2 = 7.5433048$			$P \leq 0.01$		

Rule 2: S -> T					
	T	NO_T	T_ROW	SUP_L	CONF
S	36	112	148	78.31	24.32%
NO_S	19	134	41		
T_COL	55	134	189		
SUP_R	29.10%			19.05%	
$\chi^2 = 7.5433048$			$P \leq 0.01$		

(*a) Rule: T-> S*
(b) Rule: S-> T

Table 8. Chi-square test on negative relationship

Rule 3: L -> R					
	R	NO_R	T_ROW	SUP_L	CONF
L	29	26	55	29.10%	52.73%
NO_L	77	57	134		
T_COL	106	83	189		
SUP_R	56.1%				15.34%
$p \leq 0.55128125$					

Rule 4: L -> NO_R					
	NO_R	R	T_ROW	SUP_L	CONF
L	26	29	55	29.10%	47.27%
NO_L	57	77	134		
T_COL	83	106	189		
SUP_R	43.9%				13.76%
$p \leq 0.55128125$					

(a) Rule 3: L -> R
(b) Rule 4: L -> NO_R

between the antecedent and the consequent in Chi-squared tests.

3. The relationship between minsup and minconf was not decisive. High level of minconf (70% above), in general, reflected significant relationship with either high or low minsup. Mid range of minconf mixed with either mid or high level of minsup can produce either significant or not-significant outcomes.

REFERENCES

Agrawal, R., Imielinski, T., & Swami, A. (1993). Mining Association Rules between Sets of Items in Large Databases. In *Proceedings of the ACM-SIGMOD Conf. on Management of Data* (pp. 207-216).

Agrawal R. & Srikant, R. (1994). Fast algorithms for Mining Association Rules. In *Proceedings of the 20th VLDB Conference* (pp. 487-499).

Aumann, Y. & Lindell, Y., (1999). A Statistical Theory for Quantitative Association Rules. In *Proceedings of 5th International Conference on Knowledge Discovery and Data Mining* (pp. 261-270).

Bayardo, J. & Agrawal, R. (1999). Mining the Most Interesting Rules. In *Proceedings of the 5th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining* (pp. 145-154).

Bayardo, R., Agrawal, R, & Gunopulos, D. (1999). Constraint-Based Rule Mining in Large, Dense Databases. In *Proceedings of the 15th Int'l Conf. on Data Engineering* (pp. 188-197).

Brin, S. Motwani, R. & Silverstein, R. (1997). Beyond Market Basket: Generalizing Association Rules to Correlations. In *Proceedings of ACM SIGMOD Conference,* (pp. 265-27).

Brin, S., Motwani, R., Ullman, J., & Tsur, S. (1997). Dynamic Itemset Counting and Implication Rules for Market Basket Data, In *Proceedings of ACM SIGMOD Conference on the Management of Data* (pp. 255-264).

Elder, J. (1999). Combining Estimators to Improve Performance: Bundling, Bagging, Boosting, and Bayesian Model Averaging. In *Proceedings of 5th International Conference on Knowledge Discovery and Data Mining* (pp. 237-265).

Glass, V., Hopkins, D. (1984). *Statistical Methods in Education and Psychology*. (2nd ed.) Prentice Hall, New Jersey.

Gokhale, D. & Kullback, S. (1978). *The Information in Contingency Tables*, Marcel Dekker Inc., New York.

Han, J. Pei, J., & Yin, J. Mining Frequent Patterns without Candidate Generation. (2000). In *Proceedings of the ACM SIMOD Conf on Management of Data* (1-12).

Liu B., Hsu W., & Ma Y. (1999). Pruning and Summarizing the Discovered Associations. In *Proceedings of the ACM SIGKDD Int'l Conference on Knowledge Discovery & Data Mining.*

Liu B., Hsu W., Wang K., & Chen S. (1999). Mining Interesting Knowledge Using DM-II. *In Proceedings of the ACM SIGKDD Int'l Conference on Knowledge Discovery & Data Mining.*

Mason, D., Lind, A. & Marchal, G. (1998). *STATISTICS: An Introduction*, 5th ed. Duxbury Press.

Park, S.; Chen, M-S.; & Yu, P. (1995). An Effective Hash Based Algorithm for Mining Association Rules. In *Proceedings of ACM SIGMOD Conference on the Management of Data* (pp. 175-186).

Srikant, R. & Agrawal, R. (1995). Mining Generalized Association Rules. *In Proceedings of the 21st Int'l Conf. on VLDB* (pp. 407-419).

Srikant, R., Vu, Q., & Agrawal, R. (1997). Mining Association Rules with Item Constraints, *In Proceedings of the Third Int'l Conf. on Knowledge Discovery in Databases and Data Mining* (pp. 67-73).

Tan, P., Kumar, V., & Srivastava, J., (2002). Selecting the Right Interestingness Measure for Association Patterns. In *Proceedings of 8th International Conference on Knowledge Discovery and Data Mining* (pp. 32-41).

Zaki, J., Parthasarathy, S., Ogihara, & M. Li, W. (1997). New Algorithms for Fast Discovery of Association Rules. In *Proceedings of the Third Int'l Conf. on Knowledge Discovery in Databases and Data Mining* (pp. 283-286).

KEY TERMS

Chi-Squared (χ^2) Test: A non-parametric statistical method that can be used to test independence among attributes.

Confidence: Given a rule X->Y, the confidence of the rule is the conditional probability that a customer buys itemset Y, given that he/she buys itemset X.

Frequent Itemsets: Itemsets that satisfy the minsup threshold.

Itemset: A set of zero or more items.

Market Basket Data: customer purchase data collected at the checkout counters of stores.

Minconf: The given minimal confidence level to validate the association rule.

Minsup: The given minimal support ratio for an itemset to qualify as a frequent itemset.

Support (Ratio): The percentage of transactions that contain a given itemset.

Quantization of Continuous Data for Pattern Based Rule Extraction

Andrew Hamilton-Wright
University of Guelph, Canada, & Mount Allison University, Canada

Daniel W. Stashuk
University of Waterloo, Canada

INTRODUCTION

A great deal of interesting real-world data is encountered through the analysis of continuous variables, however many of the robust tools for rule discovery and data characterization depend upon the underlying data existing in an ordinal, enumerable or discrete data domain. Tools that fall into this category include much of the current work in fuzzy logic and rough sets, as well as all forms of event-based pattern discovery tools based on probabilistic inference.

Through the application of discretization techniques, continuous data is made accessible to the analysis provided by the strong tools of discrete-valued data mining. The most common approach for discretization is quantization, in which the range of observed continuous valued data are assigned to a fixed number of quanta, each of which covers a particular portion of the range within the bounds provided by the most extreme points observed within the continuous domain. This chapter explores the effects such quantization may have, and the techniques that are available to ameliorate the negative effects of these efforts, notably fuzzy systems and rough sets.

BACKGROUND

Real-world data sets are only infrequently composed of discrete data, and any reasonable knowledge discovery approach must take into account the fact that the underlying data will be based on continuous-valued or mixed mode data. If one examines the data at the UCI Machine-Learning Repository (Newman, Hettich, Blake & Merz, 1998) one will see that many of the data sets within this group are continuous-valued; the majority of the remainder are based on measurements of continuous valued random variables that have been pre-quantized before being placed in the database.

The tools of the data mining community may be considered to fall into the following three groups:

- minimum-error-fit and other gradient descent models, such as: support vector machines (Cristianini & Shawe-Taylor, 2000; Duda, Hart & Stork, 2001; Camps-Valls, Martínez-Ramón, Rojo-Álvarez & Soria-Olivas, 2004); neural networks (Rumelhart, Hinton & Williams, 1986); and other kernel or radial-basis networks (Duda, Hart & Stork, 2001; Pham, 2006)
- Bayesian-based learning tools (Duda, Hart & Stork, 2001), including related random-variable methods such as Parzen window estimation
- statistically based pattern and knowledge discovery algorithms based on an event-based model. Into this category falls much of the work in rough sets (Grzymala-Busse, & Ziarko, 1999; Pawlak, 1982,1992; Singh & Minz, 2007; Slezak & Wroblewski, 2006), fuzzy knowledge representation (Boyen & Wehenkel, 1999; Gabrys 2004; Hathaway & Bezdek 2002; Höppner, Klawonn, Kruse & Runkler, 1999), as well as true statistical methods such as "pattern discovery" (Wang & Wong, 2003; Wong & Wang, 2003; Hamilton-Wright & Stashuk, 2005, 2006).

The methods in the last category are most affected by quantization and as such will be specifically discussed in this chapter. These algorithms function by constructing rules based on the observed association of data values among different quanta. The occurrence of a feature value within particular quanta may be considered an "event" and thereby all of the tools of information theory may be brought to bear. Without

the aggregation of data into quanta, it is not possible to generate an accurate count of event occurrence or estimate of inter-event relationships.

MAIN FOCUS

The discretization of continuous-valued data can be seen as a clustering technique in which the ranges of observed values are assigned to a limited set of *Q* cluster labels (sometimes referred to as a Borel set). The success or failure of a quantization structure may therefore be evaluated in terms of how well each of the *Q* clusters represents a homogenous and useful grouping of the underlying data.

The action of quantization is performed as a first step towards the discovery of the data topology, and therefore must frequently be undertaken without a great deal of knowledge of the underlying structure of the data. For this reason, such discretization is usually done using the information available in a single feature. Two major strategies for this are feature value *partitioning* and *quantization*.

Feature Value Partitioning

Partitioning schemes, such as those described in ID3 and C4.5 (Quinlan, 1986; 1993) as well as those used in the WEKA project (Witten & Frank, 2000) rely upon an analysis of decisions to be made within a single feature to provide a classification specific means of dividing the observed data values between labels. Quinlan (1993) provides an excellent discussion of an information-theoretic based approach to the construction of per-feature partitioning schemes in the discussion of the C4.5 classifier. In any such partitioning scheme, the placement of the partition is chosen as a means to optimize a classification decision made based on a single feature.

Quinlan's (1993) discussion is particularly salient, as it is in such tree-based classifiers that this treatment is the most advantageous, because the primary feature of a partitioning mechanism is that each feature is treated independently. This supports classification algorithms that are based on a decision tree, but does not support the discovery of multi-feature, high-order events. Furthermore, note that a classifier label value must be known in advance in order to use this technique; all data is therefore viewed in terms of its ability to provide support for some particular label value.

Feature Value Quantization

Quantization, on the other hand, refers to the construction of a set of range-based divisions of the input feature space, where each distinct quantization "bin" represents a projection of an input feature through an aggregation scheme, independent of label value. By constructing such a quantization independently of class label values, it is therefore possible to support the discovery of data patterns independent of any potential classification structure. Feature value quantization underlies most fuzzy systems (Pedrycz, 1995; Pal & Mitra, 1999) as well as discrete information theoretic based approaches such as the "pattern discovery" algorithm (Wang & Wong, 2003; Wong & Wang, 2003; Hamilton-Wright & Stashuk, 2006).

It is through the introduction of uncertainty-management techniques that the "cost of quantization" may be ameliorated. This cost is inherent in the structure of the quantization resulting from a particular technique.

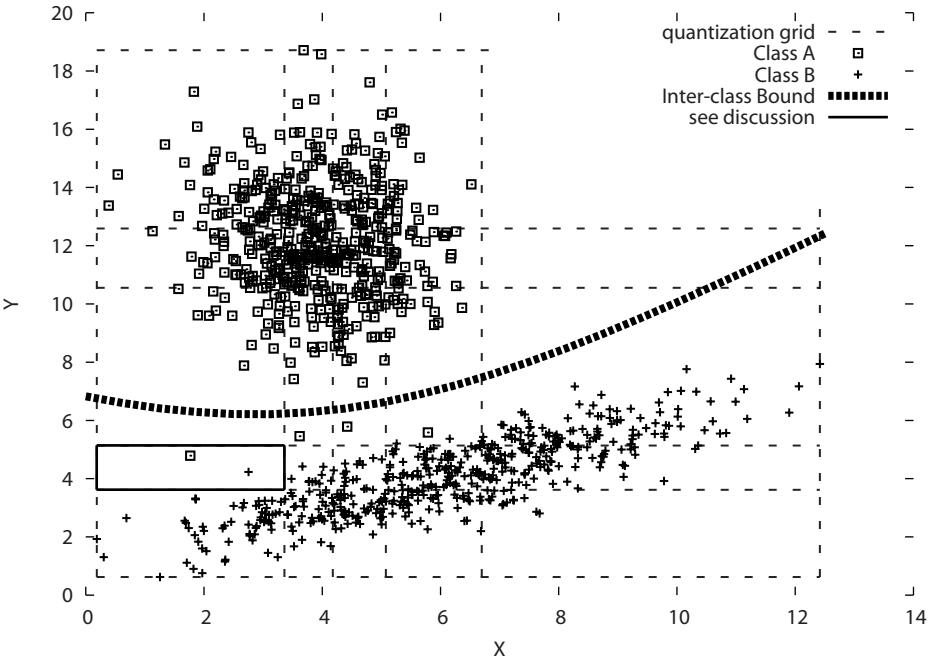
Properties of Quantization

The main strength of quantization is that by reducing a continuous domain problem to a discrete one, the powerful tools of event-based reasoning may be brought to bear. By taking a problem from the continuous domain to a discrete one, the random variables underlying the model of the continuous domain may be represented by discrete state variables, thereby allowing the representation of the data domain as a series of events with possible correlation (for an example, see Ross, 1985).

As an added benefit, such quantization provides protection against measurement noise (as analysis will be insensitive to perturbation in measurement value) provided such perturbation does not change the bin assignment. This simplifies the overall problem, reducing computational complexity, which in turn allows the application of algorithms that locate and analyze high-order patterns (*i.e.*; significant correlations between a large number of input features) such as the "pattern discovery" algorithm just mentioned.

All problems arising from the use of quantization stem from the fact that no quantization scheme can

Figure 1. Quantization example



accurately capture all of the underlying information in the continuous data representation. Consider the data in Figure 1; this data has been quantized into Q=5 quanta using the marginal maximum entropy quantization protocol (Gokhale, 1999; Chau, 2001), or "MME". The MME protocol functions by producing quantization bin boundaries that place an equal number of observed points into each bin for each feature in the input data. In Figure 1, it is clear that the data has been divided among 5 bins in each feature. The width of the bins varies inversely with point density, producing narrower bins where there is a higher density of data points, and wider bins elsewhere. This produces the optimal estimate of bin boundaries for the purpose of probabilistically based data mining; in other words, as each bin contains the same number of points, the statistical support that may be drawn from any bin is equivalent. It has been shown by Chau (2001) that this provides the optimal basis for the quantized representation of an unknown distribution topology from the point of view of observing statistical correlations between event-based bin values.

The choice of Q, the number of bins among which to divide the observed feature values, will be based on the rule discovery mechanism to be used. A mechanism to estimate the maximum possible value for Q based on event-based expectation is provided in Hamilton-Wright & Stashuk (2005); the methodology described in that work may easily be extrapolated to other statistically based classifiers.

Quantization Structure and Inter-Class Decision Boundaries

In cases where the purpose of quantization is to support prediction of a class label, MME quantization may pose a number of problems. For the data set displayed in Figure 1, using MME class-independent quantization the locations of the quantization bin boundaries are independent of the location of the optimal inter-class bound (constructed for Figure 1 as described in Duda, Hart & Stork (2001), pp. 41, for Gaussian "Normal" distributions). As such, the MME quantization bin structure shown in Figure 1 does not directly support

the discovery of the optimal inter-class bound. This is because the optimal inter-class boundary does not follow a neat straight-line path and also because the bins themselves were created without regard to class label.

In order to follow the optimal inter-class division for the data set shown all forms of regular quantization must be abandoned. If transparency (*i.e.*; the ability to directly explain classification decisions based on feature values) is not an objective, the reader is advised to turn to high-dimensional kernel methods such as support vector machines or radial-basis function based Bayesian methods (Duda, Hart & Stork, 2001) which can accurately model the optimal inter-class bound in a non-linear space, but they can not provide a direct explanation of their results in terms of feature values. Alternatively, assuming that transparency is a significant goal within the objectives of a data mining task, the transparent explanation of classification inference available through the use of a rule production scheme based on quantization may be preferable.

Imprecision in Quantization Bin Boundaries

A further issue requiring discussion is the lack of precise information regarding the construction of the bins themselves. Note that although the quantization grid has been constructed in order to place an equal number of points into each division *by feature*, this does not imply that there are an equal number of points in each multi-feature quantum. It is precisely this fact that allows us to produce an occurrence-based estimate of the importance of each quantum for purposes of rule discovery (Pawlak, 1992; Muresan, Lásló & Hirota, 2002; Tsumoto, 2002; Ziarko, 2002a,b; Wang & Wong, 2003; Hamilton-Wright & Stashuk, 2005). This irregularity drives rule production, as quanta that contain observed numbers of occurrences that are different from those expected may be directly assumed to be rules (Haberman, 1973, 1979; Gokhale, 1999). This also implies that the bounds of different quanta are defined with imperfect information.

Specific feature values used to define quantization bin boundaries. Each quantization bin boundary may be seen as a classifier in its own right. Therefore, the number of data values supporting each boundary provides a measure of precision (or imprecision) related to the definition of the bin bounds. Bin bounds supported by a small number of points are less precisely defined than those determined by a large number of points. This imprecision, or more accurately, *vagueness* in the bin bound definition leads to a desire to represent the bin bounds using one of the several mechanisms for computing with imprecise data. The two most common of these are fuzzy logic and rough sets.

Using fuzzy sets to define quantization bins and fuzzy membership for computing in event based spaces can be quite successful (Hamilton-Wright, Stashuk & Tizhoosh 2006), it is not only possible to improve a model's accuracy (as measured by classifier performance), but it is further possible to accurately represent points that cannot be represented using a crisp model.

Considering again the data shown in Figure 1, the box shown at the left of the figure directly below the optimal decision boundary contains only two observed points; one point from each class. No probabilistically based scheme for rule production will generate a rule for this for the purpose of class-based assignment, however by relaxing the bin boundaries using a fuzzy membership system, rules found for adjacent bins may be used to generate label assignments for these points at less than unity membership; in such an assignment the points will be assigned to the closest bin in the geometric system constructed through a fuzzy membership function.

FUTURE TRENDS

It is clear that the problems associated with quantization are not easily overcome. However, quantization brings with it highly attractive features, such as the high degree of transparency and the tractability of high-quality statistically based tools. Given these benefits quantization schemes are not easily abandoned.

It is therefore not surprising that a number of projects are continuing work on adaptation and application of fuzzy and/or rough sets in an effort to recover information from a quantized data domain (Singh & Minz, 2007; Slezak & Wroblewski, 2006). Further, class-dependent quantization, either by itself or in conjunction with the aforementioned works promises to provide assistance in classification problems. The major obstacle, however, remains: how does one best represent a limited number of continuous training data points so that they can best support high-level rule discovery.

CONCLUSIONS

Quantization based schemes seemingly discard important data by grouping individual values into relatively large aggregate groups; the use of fuzzy and rough set tools helps to recover a significant portion of the data lost by performing such a grouping. If quantization is to be used as the underlying method of projecting continuous data into a form usable by a discrete-valued knowledge discovery system, it is always useful to evaluate the benefits provided by including a representation of the vagueness derived from the process of constructing the quantization bins.

REFERENCES

Boyen, X. & Wehenkel, L. (1999). Automatic Induction of Fuzzy Decision Trees and its Application to Power System Security Assessment. *Fuzzy Sets & Systems.* 102(1), 3-19.

Camps-Valls, G., Martínez-Ramón M., Rojo-Álvarez, J.L. & Soria-Olivas, E. (2004). Robust γ-filter using support vector machines. *Neurocomputing*. 62, 493--499.

Chan, K.C.C., Wong, A.K.C. & Chiu, D.K.Y. (1994). *IEEE Transactions on Systems, Man and Cybernetics.* 24(10), 1532-1547.

Chau, T. (2001). Marginal maximum entropy partitioning yields asymptotically consistent probability density functions. *IEEE Transactions on Pattern Analysis & Machine Intelligence.* 23(4), 414-417.

Chau, T. & Wong, A.K.C. (1999). Pattern discovery by residual analysis and recursive partitioning. *IEEE Transactions on Knowledge & Data Engineering.* 11(6), 833-852.

Chiang, I-J. & Hsu, J.Y-J. (2002). Fuzzy classification on trees for data analysis. *Fuzzy Sets and Systems.* 130(1), 87-99.

Ching, J.Y., Wong, A.K.C & Chan, K.C.C (1995). Class-dependent discretization for inductive learning from continuous and mixed-mode data. *IEEE Transactions on Pattern Analysis & Machine Intelligence.* 17(7), 641-651.

Cristianini, N. & Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines.* Cambridge University Press.

Duda, R.O., Hart P.E. & Stork D.G. (2001). *Pattern Classification.* 2nd edition. New York: Wiley.

Gabrys, B. (2004). Learning hybrid neuro-fuzzy classifier models from data: To combine or not to combine? *Fuzzy Sets and Systems.*147(1), 39-56.

Gokhale, D.V. (1999). On joint and conditional entropies. *Entropy.* 1(2), 21-24.

Grzymala-Busse, J. & Ziarko, W. (1999). Discovery through rough set theory. *Communications of the ACM*. 42, 55-57.

Haberman, S.J. (1973). The analysis of residuals in cross-classified tables. *Biometrics.* 29(1), 205-220.

Haberman, S.J. (1979). *Analysis of Qualitative Data.* Toronto: Academic Press. pp. 78-83.

Hamilton-Wright, A. & Stashuk, D.W. (2005, July). *Comparing 'pattern discovery' and back-propagation classifiers.* In Proceedings of the International Joint Conference on Neural Networks (IJCNN'05). IEEE. Montréal, Québec.

Hamilton-Wright, A. & Stashuk, D.W. (2006). Transparent decision support using statistical reasoning and fuzzy inference. *IEEE Transactions on Knowledge & Data Engineering.* 18(8), 1125-1137.

Hamilton-Wright, A., Stashuk, D.W. & Tizhoosh, H.R. (2006). Fuzzy clustering using pattern discovery. *IEEE Transactions on Fuzzy Systems.* Accepted for publication, DOI: 10.1109/TFUZZ.2006.889930.

Hathaway, R. J. & Bezdek, J.C. (2002). Clustering incomplete relational data using the non-Euclidean relational fuzzy c-means algorithm. *Pattern Recognition Letters.* 23, 151-160.

Höppner, F., Klawonn, F., Kruse, R. & Runkler, T. A. (1999). *Fuzzy Cluster Analysis.* England: Chichester Press.

Metz, C.E. (1978). Basic principles of ROC analysis. *Seminars in Nuclear Medicine.* 8, 283-298.

Muresan, L. Lásló, K.T. & Hirota, K. (2002). *Similarity in hierarchical fuzzy rule-base systems.* In *Proceedings*

of the 11th International Conference on Fuzzy Systems. Honolulu, Hawaii.

Newman, D.J., Hettich, S., Blake, C.L. & Merz, C. J. (1998). *UCI Repository of Machine Learning Databases.* University of California, Irvine, Department of Information and Computer Sciences. http://www.ics.uci.edu/~mlearn/MLRepository.html

Pal, S. K. & Mitra, S. (1999). *Neuro-Fuzzy Pattern Recognition: Methods in Soft Computing.* Wiley Series on Intelligent Systems. Wiley-Interscience.

Pawlak, Z. (1982). Rough sets. *International Journal of Computing and Information Sciences. 11(5), 341-356.*

Pawlak, Z. (1992). *Rough Sets: Theoretical Aspects of Reasoning about Data*. Studies in Fuzziness and Soft Computing. Norwell MA: Kluwer Academic.

Pedrycz, W. (1995). *Fuzzy Sets Engineering*. CRC Press.

Pham, H. (Ed.). (2006). *Springer Handbook of Engineering Statistics.* Springer.

Rumelhart, D.E., Hinton, G.E. & Williams, R.J. (1986). Learning representations by back-propagating errors. *Nature.* 323, 533-536.

Quinlan, J. R. (1986). Induction of Decision Trees. *Machine Learning,* 1(1), 81-106.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. San Fransico: Morgan Kaufman.

Ross, S.M. (1985). *Introduction to Probability Models.* Toronto: Academic Press.

Singh, G.K. and Minz, S. (2007). *Discretization Using Clustering and Rough Set Theory.* **In** ICCTA '07, International Conference on Computing: Theory and Applications, 330-336.

Slezak, D. and Wroblewski J. (2006). *Rough Discretization of Gene Expression Data*. In ICHIT '06, International Conference on Hybrid Information Technology, (2) 265 - 267

Tsumoto, S. (2002). S*tatistical evidence for rough set analysis.* In *Proceedings of the 11th International Conference on Fuzzy Systems.* Honolulu, Hawaii.

Witten, I. H. and Frank, E. (2000). *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations.* San Francisco: Morgan Kaufman.

Wang, Y. & Wong, A.K.C. (2003). From association to classification: Inference using weight of evidence. *IEEE Transactions on Knowledge & Data Engineering.* 15(3) 764-747.

Wong, A.K.C. & Wang, Y. (2003). Pattern discovery: a data driven approach to decision support. *IEEE Transactions on Systems Man and Cybernetics C.* 33(11), 114-124.

Ziarko, W. (2002a). *Acquisition of sierarchy-structured probabalistic decision tables and rules from data.* In *Proceedings of the 11th International Conference on Fuzzy Systems.* Honolulu, Hawaii.

Ziarko, W. (2002b). *Probabalistic decision dables in the variable-precision rough-set model.* In *Proceedings of the 11th International Conference on Fuzzy Systems.* Honolulu, Hawaii.

KEY TERMS

Borel Set: A division of a data space into non-overlapping (hyper) rectangular structures; a quantization scheme frequently forms by its quanta a collection of Borel sets covering the entire data space.

Data Value Partitioning: In contrast to "data value quantization" (below) this describes a technique in which a single partition point divides the range of a feature; this is performed in support of a related decision, which is optimized according to some metric. See Quinlan (1993) for a discussion of the construction of such partitions based on mutual information; Metz (1978) provides a random-variable analysis in the context of ROC decision threshold construction.

Data Value Quantization: The division of the observed feature range among several "quantization bins" for the purpose of providing a discrete representation of the values observed in a continuous valued feature; this representation is achieved by using the numeric bin index to represent all values in the given bin.

Event Based Learning: A specific type of statistically based machine learning in which the universe of

possible input values is drawn from a set of discrete events; each event may be represented by the intersection of several discrete variables, in which case it may be described as a Borel set.

Machine Learning: An algorithm that (a) modifies its output in accordance with reduced perceived error over time by observation of a series of inputs over time, or (b) constructs an output based on likely correspondences observed over a fixed training data set; these are frequently referred to as "online" and "offline" learning schemes. See also "statistically based learning" and "event based learning".

Parzen Windows: A technique for estimating the probability density function of a random variable by interpolating based on a few sample points. See Duda, Hart & Stork (2001) for an excellent description of this technique.

Pattern Discovery: The activity of discovering patterns in a training data set for the purpose of model building and data characterization; also a particular algorithm produced for this purpose that functions by locating events whose occurrence differs significantly from a predefined hypothetical model of noise (Wang & Wong, 2003; Hamilton-Wright, Stashuk & Tizhoosh, 2007).

Rule Discovery, Rule Extraction: A form of pattern discovery in which the patterns found are intended for use in a system in which a label or course of action is suggested by an evaluation of the best matching rule or rules. Decision support systems and data classification systems are common users of discovered rules.

Statistically Based Learning: A branch of machine learning in which the underlying model of the observed data is formed through statistical or probabilistic analysis of the input data. The two main branches of this field are Bayesian inference based systems and event-based inference systems. The first are characterized by a reliance on an underlying model describing the probability density function of all of the joint probabilities associated with each decision; the second by the use of an event-based (and therefore quantized) approach to achieve discrete events that may then be counted in crisp or fuzzy terms.

Realistic Data for Testing Rule Mining Algorithms

Colin Cooper
Kings' College, UK

Michele Zito
University of Liverpool, UK

INTRODUCTION

The association rule mining (ARM) problem is a well-established topic in the field of knowledge discovery in databases. The problem addressed by ARM is to identify a set of relations (associations) in a binary valued attribute set which describe the likely coexistence of groups of attributes. To this end it is first necessary to identify sets of items that occur frequently, i.e. those subsets F of the available set of attributes I for which the *support* (the number of times F occurs in the dataset under consideration), exceeds some threshold value. Other criteria are then applied to these item-sets to generate a set of association rules, i.e. relations of the form $A \rightarrow B$, where A and B represent disjoint subsets of a frequent item-set F such that $A \cup B = F$. A vast array of algorithms and techniques has been developed to solve the ARM problem. The algorithms of Agrawal & Srikant (1994), Bajardo (1998), Brin, et al. (1997), Han *et al.* (2000), and Toivonen (1996), are only some of the best-known heuristics.

There has been recent growing interest in the class of so-called *heavy tail* statistical distributions. Distributions of this kind had been used in the past to describe word frequencies in text (Zipf, 1949), the distribution of animal species (Yule, 1925), of income (Mandelbrot, 1960), scientific citations count (Redner, 1998) and many other phenomena. They have been used recently to model various statistics of the web and other complex networks Science (Barabasi & Albert, 1999; Faloutsos *et al.*, 1999; Steyvers & Tenenbaum, 2005).

BACKGROUND

Although the ARM problem is well studied, several fundamental issues are still unsolved. In particular the evaluation and comparison of ARM algorithms is a very difficult task (Zaiane, et al., 2005), and it is often tackled by resorting to experiments carried out using data generated by the well established QUEST program from the IBM Quest Research Group (Agrawal & Srikant, 1994). The intricacy of this program makes it difficult to draw theoretical predictions on the behaviour of the various algorithms on input produced by this program. Empirical comparisons made in this way are also difficult to generalize because of the wide range of possible variation, both in the characteristics of the data (the structural characteristics of the synthetic databases generated by QUEST are governed by a dozen of interacting parameters), and in the environment in which the algorithms are being applied. It has also been noted (Brin, et al., 1997) that data sets produced using the QUEST generator might be inherently not the hardest to deal with. In fact there is evidence that suggests that the performances of some algorithms on real data are much worse than those found on synthetic data generated using QUEST (Zheng, et al., 2001).

MAIN FOCUS

The purpose of this short contribution is two-fold. First, additional arguments are provided supporting the view that real-life databases show structural properties that are very different from those of the data generated by QUEST. Second, a proposal is described for an alternative data generator that is simpler and more realistic than QUEST. The arguments are based on results described in Cooper & Zito (2007).

Heavy-Tail Distributions in Market Basket Databases

To support the claim that real market-basket databases show structural properties that are quite

different from those of the data generated by QUEST, Cooper and Zito analyzed empirically the distribution of item occurrences in four real-world retail databases widely used as test cases and publicly available from http://fimi.cs.helsinki.fi/data/. Figure 1 shows an example of such a distribution (on a log-log scale) for two of these databases. Results concerning the other two datasets are in Cooper and Zito (2007).

The authors suggest that in each case the empirical distribution may fit (over a wide range of values) a heavy-tailed distribution. Furthermore they argue that the data generated by QUEST shows quite different properties (even though it has similar size and density). When the empirical analysis mentioned above is performed on data generated by QUEST (available from the same source) the results are quite different from those obtained for real-life retail databases (see Figure 2).

Differences have been found before (Zheng *et al.*, 2001) in the transaction sizes of the real-life vs. QUEST generated databases. However some of these differences may be ironed out by a careful choice of the numerous parameters that controls the output of the QUEST generator. The results of Cooper and Zito may point to possible differences at a much deeper level.

A Closer Look at QUEST

Cooper and Zito also start a deeper theoretical investigation of the structural properties of the QUEST databases proposing a simplified version of QUEST whose mathematical properties could be effectively analyzed. As the original program, this simplified version returns two related structures: the actual database

Figure 1. Log-log plots of the real-life data sets along with the best fitting lines

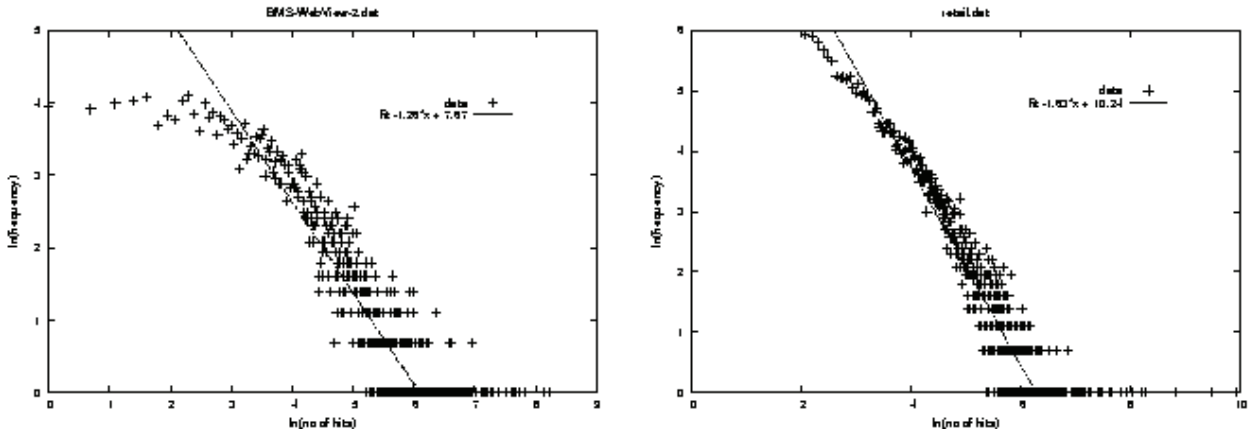
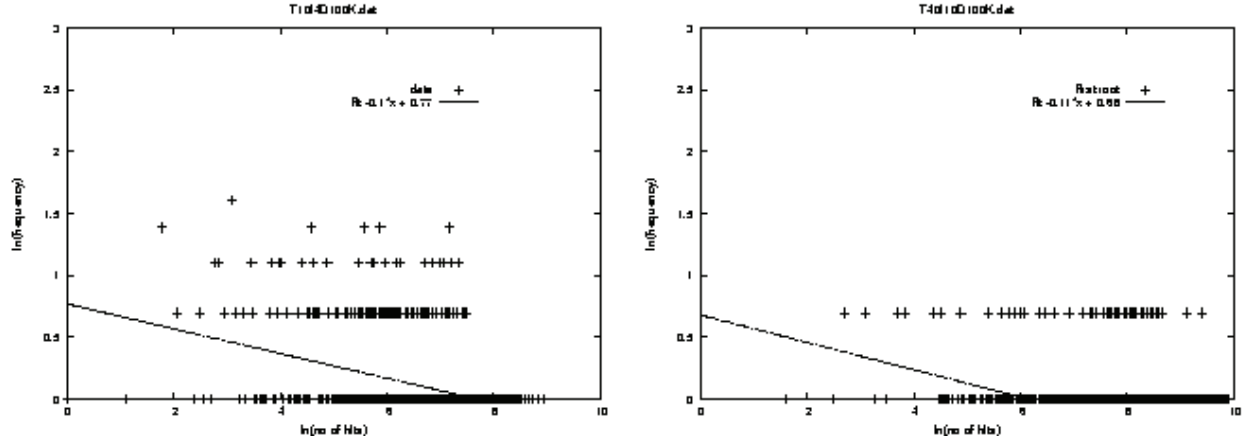


Figure 2. Log-log plots of the QUEST data sets along with the best fitting line



$\mathcal{D}$ and a collection $\mathcal{T}$ of *potentially large item-sets* (or *patterns*) that is used to populate $\mathcal{D}$. However in the simplified model, it is convenient to assume that each transaction is formed by the union of k elements of $\mathcal{T}$, chosen independently uniformly at random (with replacement). The patterns in $\mathcal{T}$ are generated by first selecting a random set of s items, and then, for each of the other patterns, by choosing (with replacement) ρ elements uniformly at random from those belonging to the last generated pattern and $s - \rho$ remaining elements uniformly at random (with replacement) from the whole set of items.

Let $\deg_D(v)$ (resp. $\deg_T(v)$) denote the number of transactions in $\mathcal{D}$ (resp. patterns in $\mathcal{T}$) containing item v. Assume that h, the total number of transactions, is a polynomial in n and $l \propto n$. It follows directly from the definition of the generation process given above that, for each item v, $\deg_D(v)$ has a binomial distribution with parameters h and $p_{k,l} = \sum_{i=1}^{k} \binom{k}{i} \frac{(-1)^{i+1}}{l^i} E(\deg_T(v)^i)$, and the expected value of N_r is $n \cdot \binom{h}{r} (p_{k,l})^r (1 - p_{k,l})^{h-r}$.

Moreover, at least in the restricted case when $s = 2$, by studying the asymptotic distribution of $\deg_T(v)$, it is possible to prove that, for constant values of k and large values of n, $p_{k,l}$ is approximately $2 \cdot n^{-1}$ and N_r is very close to its expected value. Hence for large r, the proportion of items occurring in r transaction decays much faster than r^{-z} for any fixed $z > 0$. For instance, if $k = 1$, then $\frac{N_r}{n} \to \binom{h}{r} (p_{k,l})^r (1 - p_{k,l})^{h-r}$.

An Alternative Proposal

Cooper and Zito's study of a synthetic database generator also points to possible alternatives to the IBM generator. In fact, a much more effective way of generating realistic databases is based on building the database sequentially, adding the transactions one at the time, choosing the items in each transaction based on their (current) popularity (a mechanism known as *preferential attachment*). The database model proposed by Cooper and Zito (which will be referred to as **CoZi** from now on) is in line with the proposal of Barabasi & Albert (1999), introduced to describe structures like the scientific author citation network or the world-wide web. Instead of assuming an underlying set of patterns $\mathcal{T}$ from which the transactions are built up, the elements of $\mathcal{D}$ are generated sequentially. At the start there is an initial set of e_0 transactions on n_0 existing items. **CoZi** can generate transactions based entirely on the n_0 initial items, but in general new items can also be added to newly defined transactions, so that at the end of the simulation the total number of items is $n > n_0$. The simulation proceeds for a number of steps generating groups of transactions at each step. For each group in the sequence there are four choices made by the simulation at step t:

1. The type of transaction. An `OLD` transaction (chosen with probability $1 - \alpha$) consists of items occurring in previous transactions. A `NEW` transaction (chosen with probability α) consists of a mix of new items and items occurring in previous transactions.
2. The number of transactions in a group, $m_0(t)$ (resp. $m_N(t)$) for `OLD` (resp. `NEW`) transactions. This can be a fixed value, or given any discrete distribution with mean $\overline{m_O}$ (resp. $\overline{m_N}$). Grouping corresponds to e.g. the persistence of a particular item in a group of transactions in the QUEST model.
3. The transaction size. This can again be a constant, or given by a probability distribution with mean $\overline{\pi}$.
4. The method of choosing the items in the transaction. If transactions of type `OLD` (resp. `NEW`) are chosen in a step we assume that each of them is selected using preferential attachment with probability P_0 (resp. P_N) and randomly otherwise.

The authors provide:

1. A proof that the **CoZi** model is guaranteed to generate heavy-tailed datasets, and
2. details of a simple implementation in Java, available from the authors web-sites.

More specifically, following Cooper (2006) they prove that, provided that the number of transactions is large, with probability approaching one, the distribution of item occurrence in $\mathcal{D}$ follows a power law distribution with parameter $z = 1 + \frac{1}{\eta}$, where $\eta = \frac{\alpha \overline{m_N}(\overline{\pi} - 1) P_N + (1 - \alpha)\overline{m_O}\,\overline{\pi} P_O}{(\alpha \overline{m_N} + (1 - \alpha)\overline{m_O})\,\overline{\pi}}$. In otherwords, the number of items occurring r times after t steps of the generation process is approximately Ctr^{-z} for large r and some constant $C > 0$.

Turning to examples, in the simplest case, the group sizes are fixed (say $m_N(t) = m_0(t) = 1$ always) and the preferential attachment behaviour is the same ($P_N = P_0 = P$). Thus $\eta = P - \frac{\alpha P}{\pi}$, and $z = 1 + \frac{\pi}{(\pi - \alpha P)}$. A simple implementation in Java of the **CoZi** generator, based on these settings (and the additional assumption that the transaction sizes are given by the absolute value of a normal distribution), is available at http://www.csc.liv.ac.uk/~michele/soft.html. Figure 3 displays item distribution plots obtained from running the program with P = 50% for different values of h and α.

While there are many alternative models for generating heavy tailed data (Mitzenmacher, 2004; Watts, 2004) and different communities may prefer to use alternative processes, we contend that synthetic data generators of this type should be a natural choice for the testing of ARM algorithms.

FUTURE TRENDS

Given the ever increasing needs to store and analyse large amounts of data, we anticipate that tasks like ARM or classification or pattern analysis will become increasingly important. In this context it will be desirable to have sound mathematical models that could be used to generate synthetic data or test cases. As a step in this direction, in our recent work, we looked at models for market-basket databases. Another interesting area one could look at is that of generators of classifier test cases. We believe that the theory of random processes, and, more generally, probability theory, could provide a range of techniques to devise realistic models and help discover interesting structural properties of the resulting data sets.

CONCLUSION

The association rule mining problem is a very important topic within the Data Mining research field. We provided additional evidence supporting the claim that, although a large array of algorithms and techniques exist to solve this problem, the testing of such algorithms is often done resorting to un-realistic synthetic data generators. We also put forward an alternative synthetic data generator that is simple to use, mathematically sound, and generates data that is more realistic than the one obtained from other generators.

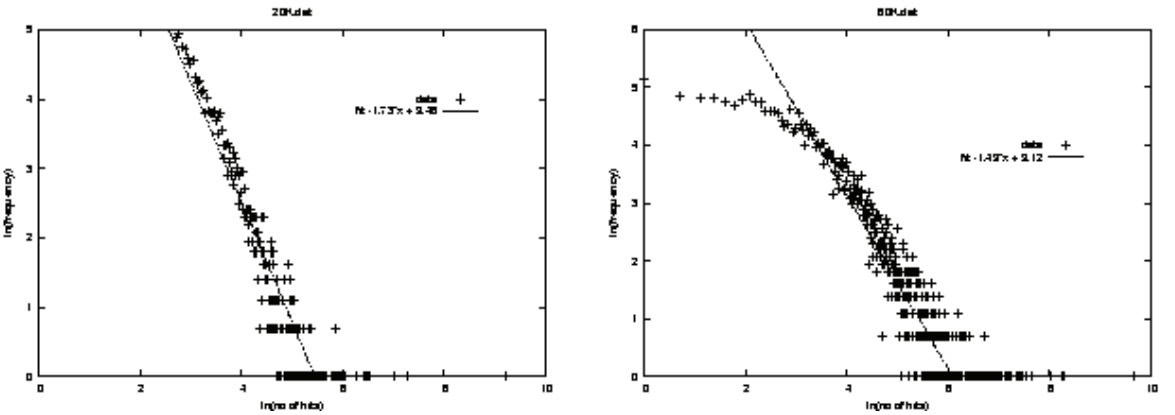
REFERENCES

Agrawal, R. & Srikant, R., (1994). Fast algorithms for mining association rules in large databases. *VLDB '94: Proceedings of the 20th International Conference on Very Large Data Bases*, San Francisco, USA, pp. 487-499.

Barabasi, A. & Albert, R., (1999). Emergence of scaling in random networks. *Science*, Vol. 286, pp. 509-512.

Bayardo, R. J. (1998). Efficiently mining long patterns from databases. *SIGMOD '98: Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, Seattle, USA, pp. 85-93.

Figure 3. Log-log plots of two CoZi data sets along with the best fitting line



Brin, S. et al, (1997). Dynamic itemset counting and implication rules for market basket data. *SIGMOD '97: Proceedings of the 1997 ACM SIGMOD International conference on Management of data*, Tucson, USA, pp. 255-264.

Cooper, C. (2006). The age specific degree distribution of web-graphs. *Combinatorics, Probability and Computing*, Vol. 15, No. 5, pp. 637-661.

Cooper, C., & Zito, M. (2007). Realistic synthetic data for testing association rule mining algorithms for market basket databases. *Knowledge Discovery in Databases: PKDD 2007*, Warsaw, Poland, *Lecture Notes in A.I.*, vol. 4702, pp. 398-405. Springer Verlag.

Faloutsos, M., Faloutsos, P., & Faloutsos, C. (1999). On power-law relationships of the internet topology. *ACM SIGCOMM Computer Communication Review*, 29(4):251-262.

Han, J. et al. (2000). Mining frequent patterns without candidate generation. *SIGMOD '00: Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, Dallas, USA, pp. 1-12.

Mandelbrot, B. (1960). The Pareto-Levy law and the distribution of income. *International Economic Review*, 1,79-106.

Mitzenmacher, M. (2004). A brief history of generative models for power-law and log-normal distributions. *Internet Mathematics*, 1(2), 226-251.

Redner, S. (1998). How popular is your paper? An empirical study of the citation distribution. *European Physical Journal*, B 4:401-404.

Steyvers, M., & Tenenbaum, J. B. (2005). The large-scale structure of semantic networks: statistical analysis and a model of semantic growth. *Cognitive Science*, 29, 41-78.

Toivonen, H. (1996). Sampling large databases for association rules. In *VLDB '96: Proceedings of the 22nd International Conference on Very Large Databases*, pages 134-145, San Francisco, USA: Morgan Kauffmann Publishers Inc.

Yule, U. (1925). A Mathematical Theory of Evolution Based on the Conclusions of Dr. J. C. Wills, F. R. S. *Philosophical Transactions of the Royal Society of London*, 213 B, 21-87.

Watts, D. J. (2004). The "new" science of networks. *Ann. Rev of Sociol.*, Vol. 30, pp. 243-270.

Zaiane, O., El-Hajj, M., Li, Y., & Luk, S. (2005). Scrutinizing frequent pattern discovery performance. In *ICDE '05: Proceedings of the 21st International Conference on Data Engineering (ICDE'05)*, pages 1109-1110, Washington DC, USA: IEEE Computer Society.

Zaki, M. J. & Ogihara, M., 1998. Theoretical foundations of association rules. *Proc. 3rd SIGMOD Workshop on Research Issues in DM and Knowl. Discov.*, Seattle, USA, pp. 1-8.

Zheng, Z. et al. (2001). Real world performance of association rule algorithms. *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, San Francisco, USA, pp. 401-406.

Zipf, G. K. (1949). *Human Behaviour and the Principle of Least Effort.* Addison-Wesley.

KEY TERMS

Association Rule Mining: The problem addressed is to identify a set of relations in a binary valued attribute set which describe the likely coexistence of groups of attributes.

Best Fitting Line: On a data diagram, this is the line drawn as near as possible to the various points so as to best represent the trend being graphed. The sums of the displacements of the points on either side of the line should be equal.

Heavy-Tailed Distribution: A statistical distribution is said to have a heavy tail if the fraction of the population up to a certain value x decays more slowly than e^{-cx}, for some $c > 0$, as x tends to infinity.

Power-Law Distribution: A statistical distribution is said to follow a power law decay if the fraction of the population up to a certain value x decays like x^{-c}, for some $c > 0$, as x tends to infinity.

Preferential Attachment: In a system of numerical quantities, randomly modified as time goes by, it is a positive feedback mechanism by which larger increases tend to accumulate on already large quantities.

Random Process: A random process is a sequence of random variables. It may be used to model the dynamics of a certain system.

Random Variable: A random variable is a numerical or otherwise measurable quantity associated with an experiment involving a random outcome.

Real-Time Face Detection and Classification for ICCTV

Brian C. Lovell
The University of Queensland, Australia

Shaokang Chen
NICTA, Australia

Ting Shan
NICTA, Australia

INTRODUCTION

Data mining is widely used in various areas such as finance, marketing, communication, web service, surveillance and security. The continuing growth in computing hardware and consumer demand has led to a rapid increase of multimedia data searching. With the rapid development of computer vision and communication techniques, real-time multimedia data mining is becoming increasingly prevalent. A motivating application is Closed-Circuit Television (CCTV) surveillance systems. However, most data mining systems mainly concentrate on text based data because of the relative mature techniques available, which are not suitable for CCTV systems. Currently, CCTV systems rely heavily on human beings to monitor screens physically. An emerging problem is that with thousands of cameras installed, it is uneconomical and impractical to hire the required numbers of people for monitoring. An Intelligent CCTV (ICCTV) system is thus required for automatically or semi-automatically monitoring the cameras.

BACKGROUND

CCTV Surveillance Systems

In recent years, the use of CCTV for surveillance has grown to an unprecedented level. Especially after the 2005 London bombings and the 2001 terrorist attack in New York, video surveillance has become part of everyday life. Hundreds of thousands of cameras have been installed in public areas all over the world, in places such as train stations, airports, car parks, Automatic Teller Machines (ATMs), vending machines and taxis. Based on the number of CCTV units on Putney High Street, it is "guesstimated" (McCahill & Norris 2002) that there are around 500,000 CCTV cameras in the London area alone and 4,000,000 cameras in the UK. This suggests that in the UK there is approximately one camera for every 14 people. However, currently there is no efficient system to fully utilize the capacity of such a huge CCTV network. Most CCTV systems rely on humans to physically monitor screens or review the stored videos. This is inefficient and makes proactive surveillance impractical. The fact that police only found activities of terrorists from the recorded videos after the attacks happened in London and New York shows that existing surveillance systems, which depend on human monitoring, are neither reliable nor timely. The need for fully automatic surveillance is pressing.

Challenges of Automatic Face Recognition on ICCTV Systems

Human tracking and face recognition is one of the key requirements for ICCTV systems. Most of the research on face recognition focuses on high quality still face images and achieves quite good results. However, automatic face recognition under CCTV conditions is still on-going research and many problems still need to be solved before it can approach the capability of the human perception system. Face recognition on CCTV is much more challenging. First, image quality of CCTV cameras is normally low. The resolution of CCTV cameras is not as high as for still cameras and the noise levels are generally higher. Second, the environment control of CCTV cameras is limited, which introduces large variations in illumination and

the viewing angle of faces. Third, there is generally a strict timing requirement for CCTV surveillance systems. Such a system should be able to perform in near real-time — detecting faces, normalizing the face images, and recognizing them.

MAIN FOCUS

Face Detection

Face detection is a necessary first step in all of the face processing systems and its performance can severely influence on the overall performance of recognition. Three main approaches are proposed for face detection: feature based, image based, and template matching.

Feature based approaches attempt to utilize some priori knowledge of human face characteristics and detect those representative features such as edges, texture, colour or motion. Edge features have been applied in face detection from the beginning (Colmenarez & Huang 1996), and several variations have been developed (Froba & Kublbeck 2002; Suzuki & Shibata 2004). Edge detection is a necessary first step for edge representation. Two edge operators that are commonly used are the Sobel Operator and Marr-Hildreth operator. Edge features can be easily detected with a very short time but are not robust for face detection in complex environments. Others have proposed texture-based approaches by detecting local facial features such as pupils, lips and eyebrows based on an observation that they are normally darker than the regions around them (Hao & Wang 2002). Color feature based face detection is derived from the fact that the skin color of different humans (even from different races) cluster very closely. Several color models are normally used, including RGB (Satoh, Nakamura & Kanade 1999), normalized RGB (Sun, Huang & Wu1998), HSI (Lee, Kim & Park 1996), YIQ (Wei & Sethi 1999), YES (Saber & Tekalp 1996), and YUV (Marques & Vilaplana 2000). In these color models, HSI is shown to be a very suitable when there is a large variation in feature colors in facial areas such as the eyes, eyebrows, and lips. Motion information is appropriate to detect faces or heads when video sequences are available (Espinosa-Duro, Faundez-Zanuy & Ortega 2004; Deng, Su, Zhou & Fu 2005). Normally frame difference analysis or moving image contour estimation is applied for face region segmentation. Recently, researchers tend to focus more on multiple feature methods which combine shape analysis, color segmentation, and motion information to locate or detect faces (Qian & Li 2000; Widjojo & Yow 2002).

The Template matching approach can be further divided into two classes: feature searching and face models. Feature searching techniques first detect the prominent facial features, such as eyes, nose, mouth, then use knowledge of face geometry to verify the existence of a face by searching for less prominent facial features (Jeng, Liao, Liu & Chern 1996). Deformable templates are generally used for face models for face detection. Yuille et al. (1989) extends the snake technique to describe features such as eyes and the mouth by a parameterized template. The snake energy comprises a combination of valley, edge, image brightness, peak, and internal energy. In Cootes and Taylor's work (1996), a point distributed model is described by a set of labeled points and Principal Component Analysis is used to define a deformable model.

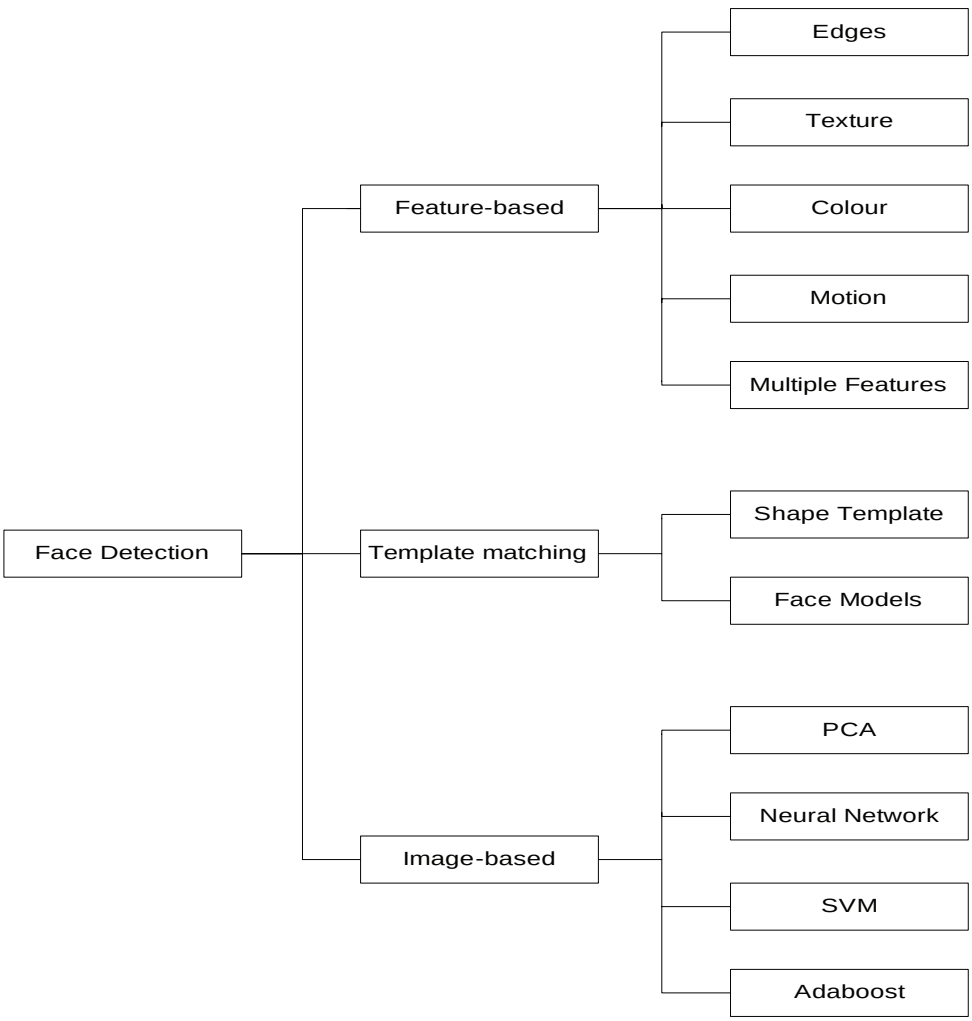
Image-based approaches treat face detection as a two class pattern recognition problem and avoid using *a priori* face knowledge. It uses positive and negative samples to train a face/non-face classifier. Various pattern classification methods are used, including Eigenfaces (Wong, Lam, Siu, & Tse 2001), Neural Network (Tivive & Bouzerdoum 2004), Support Vector Machine (Shih & Liu 2005), and Adaboost (Hayashi & Hasegawa 2006).

In summary, there are many varieties of face detection methods and to choose a suitable method is heavily application dependent. Figure 1 shows various face detection techniques and their categories. Generally speaking, feature-based methods are often used in real-time systems when color, motion, or texture information is available. Template-matching and image-based approach can attain superior detection performance than feature-based method, but most of the algorithms are computationally expensive and are difficult to apply in a real-time system.

Pose Invariant Face Recognition

Pose invariant face recognition can be classified into two categories: 2D based approaches and 3D based approaches. Although 3D face models can be used to describe the appearance of a human face under different pose changes accurately and can attain good recognition results for face images with pose variation, there

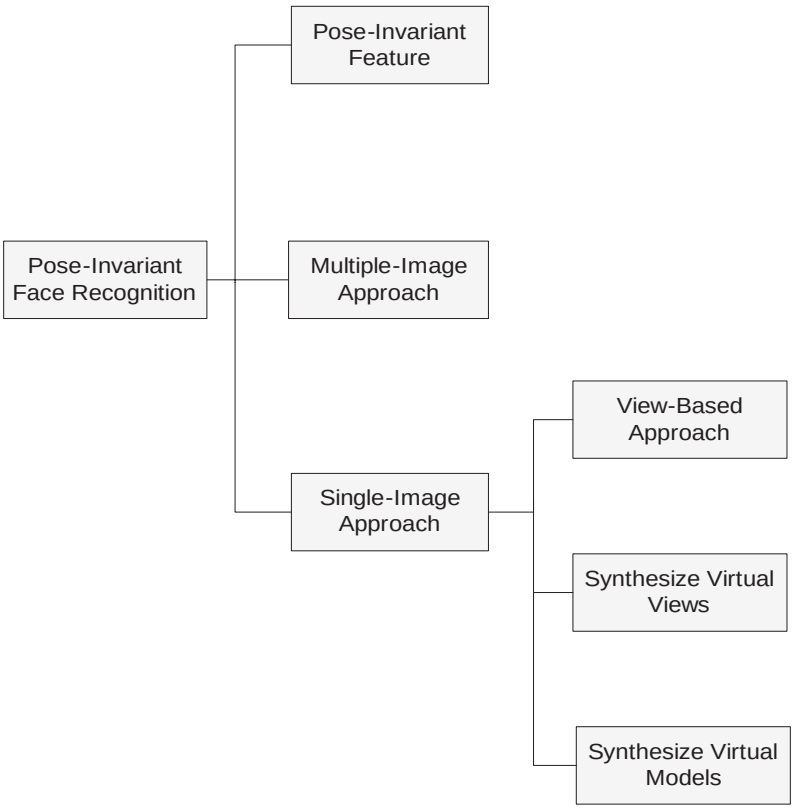
Figure 1. Various face detection techniques



are several disadvantages that limit its application to the CCTV scenario (Bowyer, Chang & Flynn 2004). First, to construct a 3D face model, 3D scanners have to be installed to replace existing cameras in the CCTV system, which is very expensive. Second, to acquire 3D data, the depth of field of the scanners has to be well controlled and this leads to the limitation on the range of data acquisition. Third, using 3D scanners to obtain 3D data is time consuming and cannot be done in real-time. Most researchers thus focus on 2D approaches for dealing with pose variation.

The 2D based approaches can be categorized into three classes: pose invariant features, multiple image, and single image as shown in Figure 2. Wiskott et al. (Wiskott, Fellous, Kuiger & Malsburg 1997) proposed Elastic Bunch Graph Matching for face recognition, which applied Gabor filters to extract pose invariant features. Beymer (1996) used multiple model views to cover different poses from the viewing sphere. Sankaran and Asari (2004) proposed a multi-view approach on Modular PCA (Pentland, Moghaddam & Starner 1994) by incorporating multiple views of the subjects as separate sets of training data. In 2001, Cootes, Edwards and Taylor (2001) proposed "View-based Active Appearance Models," based on the idea that a small number of 2D statistical models are sufficient to capture the shape and appearance of a face from any viewpoint. Sanderson *et al* (2006, 2007) addressed the pose mismatch problem by extending each frontal face model with artificially synthesized models for non-frontal views.

Figure 2. 2D based pose invariant face recognition techniques



Towards Real-Time Trials on ICCTV

The authors developed Adaptive Principal Component Analysis (APCA) to improve the robustness of PCA to nuisance factors such as lighting and expression (Chen & Lovell, 2003 and 2004). They extended APCA to deal with face recognition under variant poses (Shan, Lovell & Chen 2006) by applying an Active Appearance Model to perform pose estimation and synthesize the frontal view image. However, similar to most face recognition algorithms, the experiments were performed on some popular databases that contain only still camera images with relatively high resolution. Very few tests are done on video databases (Aggarwal 2004, Gorodnichy 2005).

We recently constructed a near real-time face detection and classification system and tested it on operational surveillance systems installed in a railway station. The system is composed of four modules: the communication module exchanges data with the surveillance system; the face detection module uses AdaBoost based cascade face detectors to detect multiple faces inside an image; the face normalization module detects facial features such as eyes and the nose to align and normalize face images; the face classification module uses the APCA method to compare face images with gallery images inside the face database. The system is implemented in C++ and runs at 7 to 10 frames per second on an Intel Dual-Core PC. Figure 3 illustrates the system structure. Our system works well in a controlled environment. If we manually align the face images of size 50 by 50 pixels or larger, it can achieves recognition rate up to 95%. But in fully automatic tests in complex uncontrolled environment, the recognition rate drops significantly. This is due to the combined effects of variations in lighting, pose, expression, registration error and image resolution etc. Figure 4 shows the real-time recognition results on two frames from a video sequence obtained from the railway station.

Figure 3. Real-time face detection and classification system structure

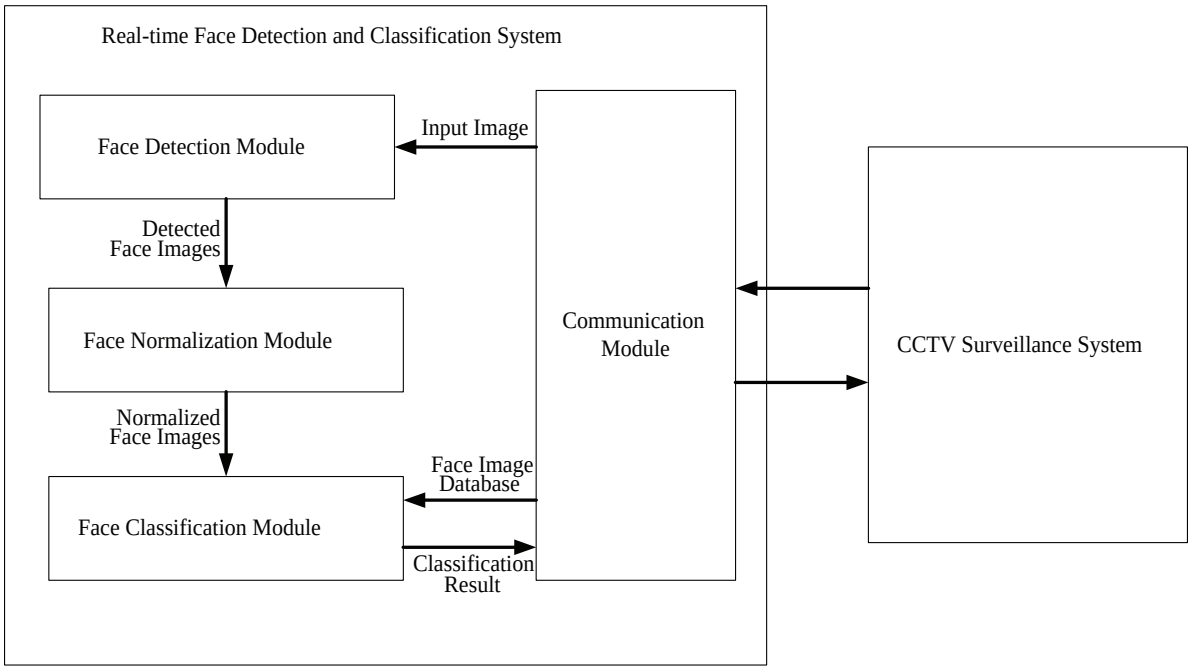


Figure 4. Real-life trial of face recognition on ICCTV. The boxes in the images show detection of the faces and the labels indicate the identity of the person from Shan et. al.(2007)

FUTURE TRENDS

Modern surveillance systems produce enormous video archives and their rate of growth is accelerating as video resolution and the number of cameras increases due to heightened security concerns. At present these archives are often erased after a few weeks due to cost of storage, but also because the archive have diminished value because there is no automatic way to search for events of interest. Face recognition provides one means to search these data for specific people. Identification reliability will be enhanced if this can be combined with, say, human gait recognition, clothing appearance models, and height information. Moreover human activity recognition could be used to detect acts of violence and suspicious patterns of behavior. Fully

integrated automatic surveillance systems are certainly the way of the future.

CONCLUSION

With the increasing demands of security, multimedia data mining techniques on CCTV, such as face detection and recognition, will deeply affect our daily lives in the near future. However, current surveillance systems which rely heavily on human operators are not practical, scalable, nor economical. This creates much interest in ICCTV systems for security applications. The challenge is that existing computer vision and pattern recognition algorithms are neither reliable nor fast enough for large database and real-time applications. But the performance and robustness of such systems will increase significantly as more attention is devoted to these problems by researchers.

REFERENCES

Aggarwal, G., Roy-Chowdhury, A.K., & Chellappa, R. (2004). A system identification approach for video-based face recognition. *Proceedings of the International Conference on Patter Recognition*, Cambridge, August 23-26.

Beymer, D., & Poggio, T. (1995). Face recognition from one example view. *Proceedings of the International Conference of Computer Vision*, 500-507.

Beymer, D. (1996). Feature correspondence by interleaving shape and texture computations. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 921-928.

Bowyer, K. W., Chang K., & Flynn P. (2004). A survey of approaches to three-dimensional face recognition. *Proceedings of the International Conference on Pattern Recognition*, *1*, 358-361.

Chen, S., & Lovell, B. C. (2003). Face recognition with one sample image per class. *Proceedings of Australian and New Zealand Intelligent Information Systems*, 83-88.

Chen, S., & Lovell, B. C. (2004). Illumination and expression invariant face recognition with one sample image. *Proceedings of the International Conference on Pattern Recognition*, *1*, 300-303.

Colmenarez, A. J., & Huang T. S. (1996). Maximum likelihood face detection. *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, 307-311.

Cootes, T. F., Edwards G. J., & C. J. Taylor (2001). Active appearance models. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, *23*(6), 681-685.

Cootes, T. F., & Taylor C. J. (1996). Locating faces using statistical feature detectors. *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, 204-209.

Deng Y-F., Su G-D., Zhou J. & Fu B. (2005). Fast and robust face detection in video. *Proceedings of the International Conference on Machine Learning and Cybernetics*, 7, 4577-4582.

Espinosa-Duro, V., Faundez-Zanuy M., & Ortega J. A. (2004). Face detection from a video camera image sequence. *Proceedings of International Carnahan Conference on Security Technology*, 318-320.

Froba, B., & Kublbeck C. (2002). Robust face detection at video frame rate based on edge orientation features. *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, 327-332.

Gorodnichy, D. (2005). Video-based framework for face recognition in video. *Proceedings of the Canadian Conference on Computer and Robot Vision*, 330-338.

Govindaraju, V. (1996). Locating human faces in photographs. *International Journal of Computer Vision*, *19*(2), 129-146.

Hao, W., & Wang K. (2002). Facial feature extraction and image-based face drawing. *Proceedings of the International Conference on Signal Processing*, 699-702.

Hayashi, S., & Hasegawa O. (2006). A detection technique for degraded face images. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1506-1512.

Jeng S-H., Liao H-Y. M., Liu Y-T., & Chern M-Y. (1996). Extraction approach for facial feature detection using geometrical face model. *Proceedings of the International Conference on Pattern Recognition*, 426-430.

Lee, C. H., Kim J. S., & Park K. H. (1996). Automatic human face location in a complex background using motion and color information. *Pattern Recognition, 29*(11), 1877-1889.

Marques, F., & Vilaplana V. (2000). A morphological approach for segmentation and tracking of human faces. *Proceedings of the International Conference on Pattern Recognition,* 1064-1067.

McCahill, M., & Norris, C. (2002). *Urbaneye: CCTV in London*. Centre for Criminology and Criminal Justice, University of Hull, UK.

Pentland, A., Moghaddam B., & Starner T. (1994). View-based and modular eigenspaces for face recognition. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 84-91.

Qian, G., & Li S. Z. (2000). Combining feature optimization into neural network based face detection. *Proceedings of the International Conference on Pattern Recognition, 2*, 814-817.

Rowley, H. A., Baluja S., & Takeo K.(1998). Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(1), 23-38.

Saber, E., & Tekalp A. M. (1996). Face detection and facial feature extraction using color, shape and symmetry-based cost functions. *Proceedings of the International Conference on Pattern Recognition, 3*, 654-658.

Sanderson, C., Bengio S., & Gao Y. (2006). On transforming statistical models for non-frontal face verification. *Pattern Recognition, 39*(2), 288-302.

Sanderson C., Shan T., & Lovell, B. C. (2007). Towards Pose-Invariant 2D Face Classification for Surveillance. *The International Workshop of Analysis and Modeling of Faces and Gestures*, 276-289.

Sankaran, P., & Asari V. (2004). A multi-view approach on modular PCA for illumination and pose invariant face recognition. *Proceedings of Applied Imagery Pattern Recognition Workshop*, 165-170.

Satoh, S., Nakamura Y., & Kanade T. (1999). Name-It: naming and detecting faces in news videos. *IEEE Multimedia*, *6*(1), 22-35.

Shan, T, Lovell, B, C., & Chen, S. (2006). Face recognition robust to head pose from one sample image. *Proceedings of the International Conference on Pattern Recognition, 1*, 515-518.

Shan, T., Chen, S., Sanderson, Sanderson C., & Lovell, B. C. (2007). Towards robust face recognition for intelligent CCTV-based surveillance using one gallery image. *IEEE International Conference on Advanced Video and Signal based Surveillance*.

Shih, P., & Liu C. (2005). Face detection using distribution-based distance and support vector machine. *Proceedings of the International Conference on Computational Intelligence and Multimedia Applications*, 327-332.

Sun, Q. B., Huang W. M., & Wu J. K. (1998). Face detection based on color and local symmetry information. *Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition*, 130-135.

Suzuki, Y., & Shibata T. (2004). Multiple-clue face detection algorithm using edge-based feature vectors. *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, *5*, V-737-40.

Tivive, F. H. C., & Bouzerdoum A. (2004). A face detection system using shunting inhibitory convolutional neural networks. *Proceedings of IEEE International Joint Conference on Neural Networks*, *4*, 2571-2575.

Viola, P., & Jones M. (2001). Rapid object detection using a boosted cascade of simple features. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *1*, I-511-I-518.

Wei, G., & Sethi I. K. (1999). Face detection for image annotation. *Pattern Recognition Letters*, *20*(11-13), 1313-1321.

Widjojo, W., & Yow K. C. (2002). A color and feature-based approach to human face detection. *Proceedings of the International Conference on Control, Automation, Robotics and Vision*, *1*, 508-513.

Wiskott, L., Fellous J. M., Kuiger N., & Malsburg von der C. (1997). Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*(7), 775-779.

Wong K. W., Lam K. M., Siu W. C., & Tse K. M. (2001). Face segmentation and facial feature tracking for videophone applications. *Proceedings of the*

International Symposium on Intelligent Multimedia, Video and Speech Processing, 518-521.

Yang M. H., Kriegman, D.J., & Ahuja, N. (2002) Detecting faces in images: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(1), 34-58.

Yuille, A. L., Cohen D. S., & Hallinan P. W. (1989). Feature extraction from faces using deformable templates. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 104-109.

KEY TERMS

Active Appearance Model: Active appearance model (AAM) is a method to represent the shape and appearance variations of objects. A statistical shape model and an appearance model are learned from landmarks of the sample data and then correlated by applying Principal Component Analysis on the training data.

Adaboost: Adaboost is a short for Adaptive Boosting. It is a boosting learning algorithm to combine many weak classifiers into a single powerful classifier. It adaptively changes the weight for weak classifiers from the misclassified data sample. It is mainly used to fuse many binary classifiers into a strong classifier.

Close-Circuit Television: Close-Circuit Television (CCTV) is a system that transmits signals from video cameras to specific end users, such as monitors and servers. Signals are transmitted securely and privately over the network. CCTV is mainly used for surveillance applications.

Face Detection: To determine whether or not there are any faces in the image and return the corresponding location of each face.

Intelligent CCTV: Intelligent CCTV (ICCTV) is a CCTV system that can automatically or semi-automatically implement surveillance functions. Some advanced techniques of computer vision, pattern recognition, data mining, artificial intelligence and communications etc. are applied into CCTV to make it smart.

Pose: In face recognition, pose means the orientation of the face in 3D space including head tilt, and rotation.

Pose Estimation: To estimate the head pose orientation in 3D space. Normally estimate angles of the head tilt and rotation.

Reasoning about Frequent Patterns with Negation

Marzena Kryszkiewicz
Warsaw University of Technology, Poland

INTRODUCTION

Discovering of frequent patterns in large databases is an important data mining problem. The problem was introduced in (Agrawal, Imielinski & Swami, 1993) for a sales transaction database. Frequent patterns were defined there as sets of items that are purchased together frequently. Frequent patterns are commonly used for building association rules. For example, an association rule may state that 80% of customers who buy fish also buy white wine. This rule is derivable from the fact that fish occurs in 5% of sales transactions and set {fish, white wine} occurs in 4% of transactions. Patterns and association rules can be generalized by admitting negation. A sample association rule with negation could state that 75% of customers who buy coke also buy chips and neither beer nor milk. The knowledge of this kind is important not only for sales managers, but also in medical areas (Tsumoto, 2002). Admitting negation in patterns usually results in an abundance of mined patterns, which makes analysis of the discovered knowledge infeasible. It is thus preferable to discover and store a possibly small fraction of patterns, from which one can derive all other significant patterns when required. In this chapter, we introduce first lossless representations of frequent patterns with negation.

BACKGROUND

Let us analyze sample transactional database D presented in Table 1, which we will use throughout the chapter. Each row in this database reports items that were purchased by a customer during a single visit to a supermarket.

As follows from Table 1, items *a* and *b* were purchased together in four transactions. The number of transactions in which set of items $\{x_1, ..., x_n\}$ occurs is called its *support* and denoted by $sup(\{x_1, ..., x_n\})$. A set of items is called a *frequent pattern* if its support exceeds a user-specified threshold (*minSup*). Otherwise, it is called an *infrequent pattern*. In the remainder of the chapter, we assume *minSup* = 1. One can discover 27 frequent patterns from D, which we list in Figure 1.

One can easily note that the support of a pattern never exceeds the supports of its subsets. Hence, subsets of a frequent pattern are also frequent, and supersets of an infrequent pattern are infrequent.

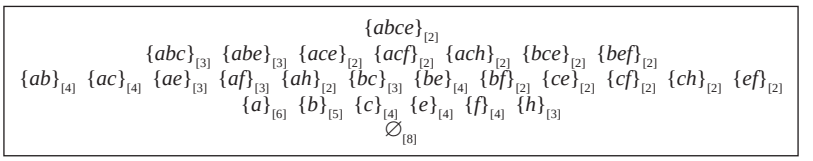
Aside from searching for only statistically significant sets of items, one may be interested in identifying frequent cases when purchase of some items (presence of some symptoms) excludes purchase of other items (presence of other symptoms). A pattern consisting of items $x_1, \ldots, x_m$ and negations of items $x_{m+1}, \ldots, x_n$ will be denoted by $\{x_1, \ldots, x_m, -x_{m+1}, \ldots, -x_n\}$. The *support of pattern* $\{x_1, \ldots, x_m, -x_{m+1}, \ldots, -x_n\}$ is defined as the number of transactions in which all items in set $\{x_1, \ldots, x_m\}$ occur and no item in set $\{x_{m+1}, \ldots, x_n\}$) occurs. In particular, $\{a(-b)\}$ is supported by two transactions in D, while $\{a(-b)(-c)\}$ is supported by one transaction. Hence, $\{a(-b)\}$ is frequent, while $\{a(-b)(-c)\}$ is infrequent.

From now on, we will say that *X* is a *positive pattern*, if *X* does not contain any negated item. Otherwise, *X* is called a *pattern with negation*. A pattern obtained from pattern *X* by negating an arbitrary number of items in *X* is called a *variation of X*. For example, {*ab*} has four

Table 1. Sample database D

Id	Transaction
T_1	{*abce*}
T_2	{*abcef*}
T_3	{*abch*}
T_4	{*abe*}
T_5	{*acfh*}
T_6	{*bef*}
T_7	{*h*}
T_8	{*af*}

Figure 1. Frequent positive patterns discovered from database D. Values provided in square brackets in the subscript denote supports of patterns.



distinct variations (including itself): {*ab*}, {*a*(–*b*)}, {(–*a*)*b*}, {(–*a*)(–*b*)}.

One can discover 109 frequent patterns in D, 27 of which are positive, and 82 of which have negated items. In practice, the number of frequent patterns with negation is by orders of magnitude greater than the number of frequent positive patterns.

A first trial to solve the problem of large number of frequent patterns with negation was undertaken by Toivonen (1996), who proposed a method for using supports of positive patterns to derive supports of patterns with negation. The method is based on the observation that for any pattern X and any item x, the number of transactions in which X occurs is the sum of the number of transactions in which X occurs with x and the number of transactions in which X occurs without x. In other words, $sup(X) = sup(X \cup \{x\}) + sup(X \cup \{(-x)\})$, or $sup(X \cup \{(-x)\}) = sup(X) - sup(X \cup \{x\})$ (Mannila and Toivonen, 1996). Multiple usage of this property enables determination of the supports of patterns with an arbitrary number of negated items based on the supports of positive patterns. For example, the support of pattern {*a*(–*b*)(–*c*)}, which has two negated items, can be calculated as follows: $sup(\{a(-b)(-c)\}) = sup(\{a(-b)\}) - sup(\{a(-b)c\})$. Thus, the task of calculating the support of {*a*(–*b*)(–*c*)}, which has two negated items, becomes a task of calculating the supports of patterns {*a*(–*b*)} and {*a*(–*b*)*c*}, each of which contains only one negated item. We note that $sup(\{a(-b)\}) = sup(\{a\}) - sup(\{ab\})$, and $sup(\{a(-b)c\}) = sup(\{ac\}) - sup(\{abc\})$. Eventually, we obtain: $sup(\{a(-b)(-c)\}) = sup(\{a\}) - sup(\{ab\}) - sup(\{ac\}) + sup(\{abc\})$. The support of {*a*(–*b*)(–*c*)} is hence determinable from the supports of {*abc*} and its proper subsets.

It was proved in (Toivonen, 1996) that for any pattern with negation its support is determinable from the supports of positive patterns. Nevertheless, the knowledge of the supports of only frequent positive patterns may be insufficient to derive the supports of all patterns with negation (Boulicaut, Bykowski & Jeudy, 2000), which we illustrate beneath.

Let us try to calculate the support of pattern $\{bef(-h)\}$: $sup(\{bef(-h)\}) = sup(\{bef\}) - sup(\{befh\})$. Pattern {*bef*} is frequent (see Figure 1), so it is stored altogether with its support. To the contrary, {*befh*} is not frequent, so the information about {*befh*} and its support is not stored. Thus, we are unable to calculate the support of {*bef*(–*h*)} based on the frequent positive patterns.

The problem of large amount of mined frequent patterns is widely recognized. Within the last decade, a number of lossless representations of frequent positive patterns have been proposed. Frequent closed itemsets were introduced in (Pasquier et al., 1999); the generators representation was introduced in (Kryszkiewicz, 2001). Other lossless representations are based on disjunction-free sets (Bykowski & Rigotti, 2001), disjunction-free generators (Kryszkiewicz, 2001), generalized disjunction-free generators (Kryszkiewicz & Gajek, 2002), generalized disjunction-free sets (Kryszkiewicz, 2003), non-derivable itemsets (Calders & Goethals, 2002), and *k*-free sets (Calders & Goethals, 2003). All these models allow distinguishing between frequent and infrequent positive patterns and enable determination of supports for all frequent positive patterns. Although the research on concise representations of frequent positive patterns is advanced, there are few papers in the literature devoted to representing of all frequent patterns with negation.

MAIN THRUST OF THE CHAPTER

We define a *generalized disjunction-free literal set model* (GDFLR) as a concise lossless representation of all frequent positive patterns and all frequent patterns with negation. Without the need to access the database, GDFLR enables distinguishing between all frequent and infrequent patterns, and enables calculation of the supports for all frequent patterns. We also define a *k-generalized disjunction-free literal set model* (*k*-GDFLR) as a modification of GDFLR for more concise lossless representing of all frequent positive patterns and all frequent patterns with at most *k* negated items.

Both representations may use the mechanism of deriving supports of positive patterns that was proposed in (Kryszkiewicz & Gajek, 2002). Hence, we first recall this mechanism. Then we examine how to use it to derive the supports of patterns with negation and introduce a respective naive representation of frequent patterns. Next we examine relationships between specific patterns and supports of their variations. Eventually, we use the obtained results to offer GDFLR as a refined version of the naive model, and *k*-GFLDR as a generalization of GDFLR which coincides with GDFLR for $k = \infty$.

Reasoning about Positive Patterns Based on Generalized Disjunctive Patterns

Let us observe that whenever item *a* occurs in a transaction in database D, then item *b*, or *f*, or both also occur in the transaction. This fact related to pattern {*abf*} can be expressed in the form of implication $a \Rightarrow b \vee f$. Now, without accessing the database, we can derive additional implications, such as $ac \Rightarrow b \vee f$ and $a \Rightarrow b \vee f \vee c$, which are related to supersets of {*abf*}. The knowledge of such implications can be used for calculating the supports of patterns they relate to. For example, $ac \Rightarrow b \vee f$ implies that the number of transactions in which {*ac*} occurs equals the number of transactions in which {*ac*} occurs with *b* plus the number of transactions in which {*ac*} occurs with *f* minus the number of transactions in which {*ac*} occurs both with *b* and *f*. In other words, *sup*({*ac*}) = *sup*({*acb*}) + *sup*({*acf*}) – *sup*({*acbf*}). Hence, *sup*({*abcf*}) = *sup*({*abc*}) + *sup*({*acf*}) – *sup*({*ac*}), which means that the support of pattern {*abcf*} is determinable from the supports of its proper subsets. In general, if there is an implication related to a positive pattern, then the support of this pattern is derivable from the supports of its proper subsets (please, see (Kryszkiewicz & Gajek, 2002) for proof). If there is such an implication for a pattern, then the pattern is called a *generalized disjunctive set*. Otherwise, it is called a *generalized disjunction-free set*. We will present now a lossless *generalized disjunction-free set representation* (GDFSR) of all frequent positive patterns, which uses the discussed mechanism of deriving supports. The GDFSR representation is defined as consisting of the following components (Kryszkiewicz, 2003):

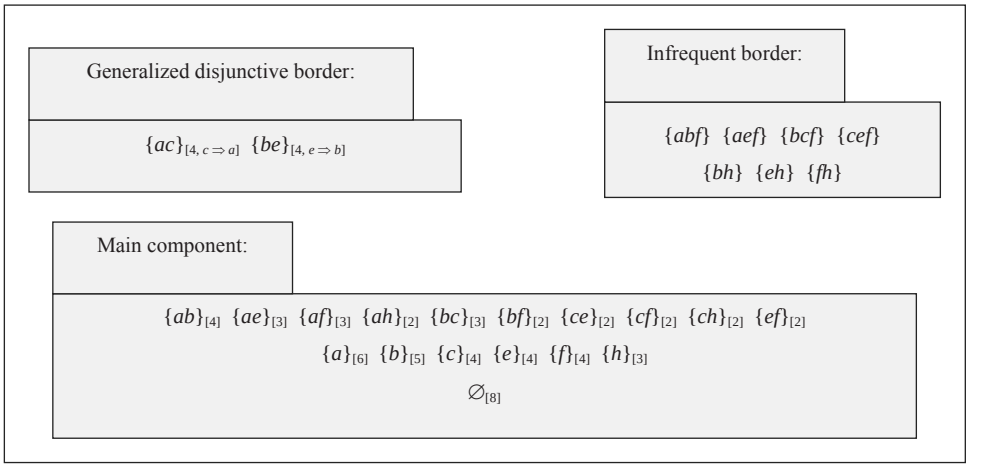
- The main component containing all frequent generalized disjunction-free positive patterns stored altogether with their supports;
- The infrequent border consisting of all infrequent positive patterns all proper subsets of which belong to the main component;
- The generalized disjunctive border consisting of all minimal frequent generalized disjunctive positive patterns stored altogether with their supports and/or respective implications.

Figure 2 depicts the GDFSR representation found in D. The main component consists of 17 elements, the infrequent border of 7 elements, and generalized disjunctive border of 2 elements.

Now, we will demonstrate how to use this representation for evaluating unknown positive patterns:

- Let us consider pattern {*abcf*}. We note that {*abcf*} has a subset, e.g. {*abf*}, in the infrequent border. This means that all supersets of {*abf*}, in particular {*abcf*}, are infrequent.
- Let us consider pattern {*abce*}. It does not have any subset in the infrequent border, but has a subset, e.g. {*ac*}, in the generalized disjunctive border. Property $c \Rightarrow a$, which is associated with {*ac*} implies property $bce \Rightarrow a$ related to {*abce*}. Hence, *sup*({*abce*}) = *sup*({*bce*}). Now, we need to determine the support of {*bce*}. We observe that {*bce*} has subset {*be*} in the generalized disjunctive border. Property $e \Rightarrow b$ associated with {*be*} implies property $ce \Rightarrow b$ related to {*bce*}. Hence, *sup*({*bce*}) = *sup*({*ce*}). Pattern {*ce*} belongs to the main component, so its support is known (here: equals 2). Summarizing, *sup*({*abce*}) = *sup*({*bce*}) = *sup*({*ce*}) = 2.

Figure 2. The GDFSR representation found in D



Naive Approach to Reasoning About Patterns with Negation based on Generalized Disjunctive Patterns

One can easily note that implications, we were looking for positive patterns, may exist also for patterns with negation. For instance, looking at Table 1, we observe that whenever item *a* occurs in a transaction, then item *b* occurs in the transaction and/or item *e* is missing in the transaction. This fact related to pattern $\{ab(\neg e)\}$ can be expressed as implication $a \Rightarrow b \vee (\neg e)$. Hence, $sup(\{a\}) = sup(\{ab\}) + sup(\{a(\neg e)\}) - sup(\{ab(\neg e)\})$, or $sup(\{ab(\neg e)\}) = sup(\{ab\}) + sup(\{a(\neg e)\}) - sup(\{a\})$. Thus, the support of pattern $\{ab(\neg e)\}$ is determinable from the supports of its proper subsets. In general, the support of a generalized disjunctive pattern with any number of negated items is determinable from the supports of its proper subsets.

Having this in mind, we conclude that the GDFSR model can easily be adapted for representing all frequent patterns. We define a *generalized disjunction-free set representation of frequent patterns admitting negation* (GDFSRN) as holding all conditions that are held by GDFSR except for the condition restricting the representation's elements to positive patterns. GDFSRN discovered from database D consists of 113 elements. It contains both positive patterns and patterns with negation. For instance, $\{bc\}_{[3]}$, $\{b(\neg c)\}_{[2]}$, and $\{(\neg b)(\neg c)\}_{[2]}$, which are frequent generalized disjunction-free, are sample elements of the main component of this representation, whereas $\{a(\neg c)\}_{[2,\ \varnothing \Rightarrow a \vee (\neg c)]}$, which is a minimal frequent generalized disjunctive pattern, is a sample element of the generalized disjunctive border. Although conceptually straightforward, the representation is not concise, since its cardinality (113) is comparable with the number of all frequent patterns (109).

Generalized Disjunctive Patterns vs. Supports of Variations

Let us consider implication $a \Rightarrow b \vee f$, which holds in our database. The statement that whenever item *a* occurs in a transaction, then item *b* and/or item *f* also occurs in the transaction is equivalent to the statement that there is no transaction in which *a* occurs without both *b* and *f*. Therefore, we conclude that implication $a \Rightarrow b \vee f$ is equivalent to statement $sup(a(\neg b)(\neg f)) = 0$. We generalize this observation as follows:

$x_1 \ldots x_m \Rightarrow x_{m+1} \vee \ldots \vee x_n$ is equivalent to $sup(\{x_1, \ldots, x_m\} \cup \{\neg x_{m+1}, \ldots, \neg x_n\}) = 0$.

Let us recall that $x_1 \ldots x_m \Rightarrow x_{m+1} \vee \ldots \vee x_n$ implies that pattern $\{x_1, \ldots, x_n\}$ is generalized disjunctive, and $sup(\{x_1, \ldots, x_m\} \cup \{\neg x_{m+1}, \ldots, \neg x_n\}) = 0$ implies that pattern $\{x_1, \ldots, x_n\}$ has a variation different from itself that does not occur in any transaction. Hence, we infer that a positive pattern is generalized disjunctive if and only if it has a variation with negation the support of which equals 0.

Effective Approach to Reasoning About Patterns with Negation based on Generalized Disjunctive Patterns

In order to overcome the problem of possible small conciseness ratio of the GDFSRN model, we offer a new representation of frequent patterns with negation. Our intention is to store in the new representation at most one pattern for a number of patterns occurring in GDFSRN that have the same positive variation.

We define a *generalized disjunction-free literal representation* (GDFLR) as consisting of the following components:

- the main component containing each positive pattern (stored with its support) that has at least one frequent variation and all its variations have non-zero supports;
- the infrequent border containing each minimal positive pattern all variations of which are infrequent and all proper subsets of which belong to the main component;
- the generalized disjunctive border containing each minimal positive pattern (stored with its support and, eventually, implication) that has at least one frequent variation and at least one variation with zero support.

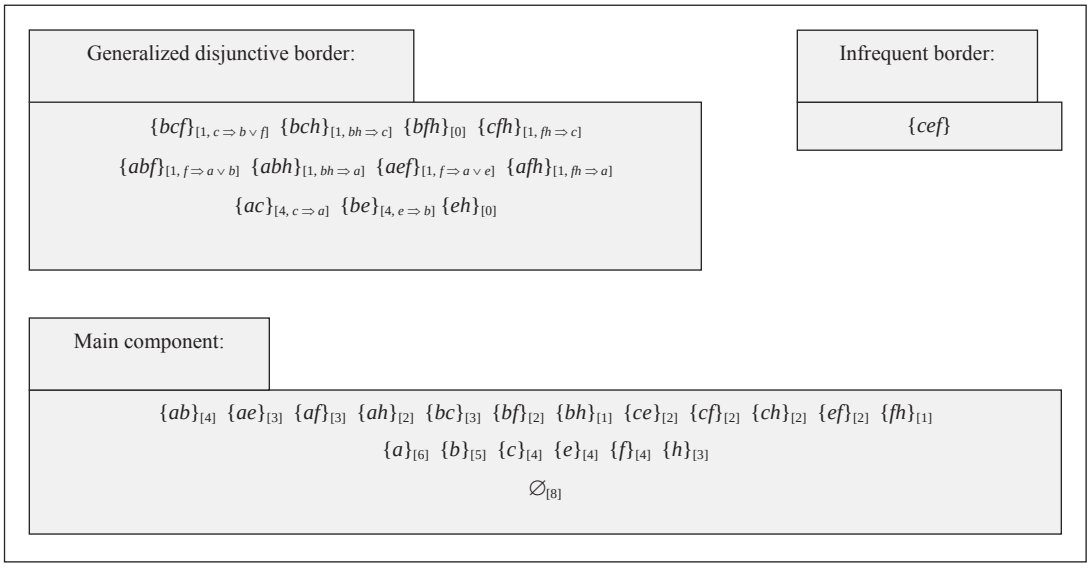
Please note that each element in the main component is generalized disjunction-free since all its variations have non-zero supports. On the other hand, each element in the generalized disjunctive border is generalized disjunctive or has support equal to zero.

Figure 3 depicts GDFLR discovered in D. The main component consists of 19 elements, the infrequent border of 1 element, and generalized disjunctive border of 11 elements.

Now we will illustrate how to use this representation for evaluating unknown patterns:

- Let us consider pattern $\{a(\neg c)(\neg e)f\}$. We note that $\{acef\}$, which is a positive variation of the evaluated pattern, has subset $\{cef\}$ in the infrequent border. This means that both $\{cef\}$ and all its variations including $\{(\neg c)(\neg e)f\}$ are infrequent. Hence, $\{a(\neg c)(\neg e)f\}$, which is a superset of $\{(\neg c)(\neg e)f\}$, is also infrequent.
- Let us consider pattern $\{bef(\neg h)\}$. The positive variation $\{befh\}$ of $\{bef(\neg h)\}$ does not have any subset in the infrequent border, so $\{bef(\neg h)\}$ has a chance to be frequent. Since, $sup(\{bef(\neg h)\}) = sup(\{bef\}) - sup(\{befh\})$, we need to determine the supports of two positive patterns $\{bef\}$ and $\{befh\}$. $\{bef\}$ has subset $\{be\}$ in the generalized disjunctive border, the implication of which is $e \Rightarrow b$. Hence, $ef \Rightarrow b$ is an implication for $\{bef\}$.

Figure 3. The GDFLR representation found in D



Thus, $sup(bef) = sup(ef) = 2$ (please, see the main component for pattern $\{ef\}$). Pattern $\{befh\}$ also has a subset, e.g. $\{eh\}$, in the generalized disjunctive border. Since $sup(\{eh\}) = 0$, then $sup(\{befh\})$ equals 0 too. Summarizing, $sup(\{bef(-h)\}) = 2 - 0 = 2$, and thus $\{bef(-h)\}$ is a frequent pattern.

GDFLR is a lossless representation of all frequent patterns. It can be proved that a pessimistic estimation of the length of a longest element in GDFLR depends logarithmically on the number of records in the database. A formal presentation of this model and its properties, as well as an algorithm for its discovery and experimental results can be found in (Kryszkiewicz, 2004b). The experiments carried out on real large data sets show that GDFLR is by several orders of magnitude more concise than all frequent patterns. Further reduction of GDFLR (and GDFSRN) can be achieved by applying techniques for reducing borders (Calders & Goethals, 2003; Kryszkiewicz, 2003; Kryszkiewicz, 2004a) or a main component (Kryszkiewicz, 2004c).

In (Kryszkiewicz & Cichon, 2005), we discuss the complexity of evaluating candidate elements of the representation. We observe that the calculation of the support of a pattern with n negated items from the supports of positive patterns requires the knowledge of the support of the positive variant P of that pattern and the supports of $2^n - 1$ proper subsets of pattern P. In order to alleviate this problem, we offer a special ordering of candidate elements in (Kryszkiewicz & Cichon, 2005). The introduced ordering enables calculation of the support of a pattern with n negated items as the difference of the supports of only two patterns, possibly with negation, the supports of which were calculated earlier. The proposed algorithm using this method of calculating supports performs much faster (by up to two orders of magnitude for low support threshold values). Nevertheless, in each iteration l, it requires storing all variants of all candidates of length l and all variants of all elements of the main component of length l-1.

Reasoning about Patterns with at Most k-Negated Items

An important part of data mining is discovering patterns conforming user-specified constraints. It is natural to expect that the derivation of a part of all frequent patterns instead of all of them should take less time and should produce less number of patterns than unrestricted data mining. One can also anticipate that a representation of a part of all frequent patterns should be more concise than the representation of all frequent patterns. In this section, we define a generalized disjunction-free literal representation of frequent patterns with at most k negated items (k-GDFLR). The new representation consists of the following components (Kryszkiewicz, 2006):

- the main component containing each positive pattern (stored with its support) that has at least one frequent variation with at most k negated items and is neither generalized disjunctive nor has support equal 0;
- the infrequent border containing each positive pattern whose all variations with at most k negated items are infrequent and whose all proper subsets belong to the main component;
- the generalized disjunctive border containing each minimal positive pattern (stored with its support and/or implication) that has at least one frequent variation with at most k negated items and at least one variation with zero support.

Please note that for $k = 0$ (no negation allowed) k-GDFLR represents all frequent positive patterns, whereas for $k = \infty$ (unrestricted number of negations) it represents all frequent patterns: both positive ones and with negation.

It was proved in (Kryszkiewicz, 2006) that the k-GDFLR representation losslessly represents all frequent patterns with at most k negated items. The conducted experiments show that k-GDFLR is more concise and faster computable than GDFLR especially for low support values and low k.

FUTURE TRENDS

Development of different representations of frequent patterns with negation and algorithms for their discovery can be considered as a short-term trend. As a long-term trend, we envisage development of representations of patterns satisfying user imposed constraints and representations of other kinds of knowledge admitting negation, such as association rules, episodes, sequential patterns and classifiers. Such research should stimulate positively the development of inductive databases, where queries including negation are common.

CONCLUSION

The set of all positive patterns can be treated as a lossless representation of all frequent patterns, nevertheless it is not concise. On the other hand, the set of all frequent positive patterns neither guarantees derivation of all frequent patterns with negation, nor is concise in practice. The GDFSRN and GDFLR representations, we proposed, are first lossless representations of both all frequent positive patterns and patterns with negation. GDFLR consists of a subset of only positive patterns and hence is more concise than analogous GDFSRN, which admits the storage of many patterns having the same positive variation. We have proposed the *k*-GDFLR representation for even more concise, lossless representing of all frequent patterns with at most *k* negated items. In (Kryszkiewicz & Cichon, 2005), we have offered a method for fast calculation of the supports of the candidate elements of the generalized disjunction-free representations, which is based on a special ordering of the candidate patterns.

REFERENCES

Agrawal, R., Imielinski, R., & Swami, A. N. (1993, May). Mining association rules between sets of items in large databases. *ACM SIGMOD International Conference on Management of Data*, Washington, USA, 207-216.

Boulicaut, J.-F., Bykowski, A., & Jeudy, B. (2000, October). Towards the tractable discovery of association rules with negations. *International Conference on Flexible Query Answering Systems*, FQAS'00, Warsaw, Poland, 425-434.

Bykowski, A., & Rigotti, C. (2001, May). A condensed representation to find patterns. *ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, PODS'01, Santa Barbara, USA, 267–273.

Calders, T., & Goethals, B. (2002, August). Mining all non-derivable frequent itemsets. *European Conference on Principles and Practice of Knowledge Discovery in Databases*, PKDD'02, Helsinki, Finland, 74-85.

Calders, T., & Goethals, B. (2003, September). Minimal k-free representations. *European Conference on Principles and Practice of Knowledge Discovery in Databases*, PKDD'03, Cavtat-Dubrovnik, Croatia, 71-82.

Kryszkiewicz, M. (2001, November-December). Concise representation of frequent patterns based on disjunction–free generators. *IEEE International Conference on Data Mining*, ICDM'01, San Jose, USA, 305–312.

Kryszkiewicz, M., (2003, July). Reducing infrequent borders of downward complete representations of frequent patterns. *Symposium on Databases, Data Warehousing and Knowledge Discovery*, DDWKD'03, Baden-Baden, Germany, 29–42.

Kryszkiewicz, M. (2004a, March). Reducing borders of *k*-disjunction free representations of frequent patterns. *ACM Symposium on Applied Computing*, SAC'04, Nikosia, Cyprus, 559-563.

Kryszkiewicz, M. (2004b, May). Generalized disjunction-free representation of frequent patterns with negation. ICS Research Report 9, Warsaw University of Technology; extended version published in *Journal of Experimental and Theoretical Artificial Intelligence*, *17*(1-2), 63-82.

Kryszkiewicz, M. (2004c, July). Reducing main components of *k*-disjunction free representations of frequent patterns. *International Conference in Information Processing and Management of Uncertainty in Knowledge-Based Systems*, IPMU'04, Perugia, Italy, 1751-1758.

Kryszkiewicz, M. (2006, April). Generalized disjunction-free representation of frequents patterns with at most *k* negations. *Advances in Knowledge Discovery and Data Mining, 10th Pacific-Asia Conference*, PAKDD'06, Singapore, 468-472.

Kryszkiewicz, M., & Cichon, K. (2005, May). Support oriented discovery of generalized disjunction-free representation of frequent patterns with negation. *Advances in Knowledge Discovery and Data Mining*, PAKDD'05, Hanoi, Vietnam, 672-682

Kryszkiewicz, M., & Gajek, M. (2002, May). Concise representation of frequent patterns based on generalized disjunction–free generators. *Advances in Knowledge Discovery and Data Mining, Pacific-Asia Conference*, PAKDD'02, Taipei, Taiwan, 159–171.

Mannila, H., & Toivonen, H. (1996, August). Multiple uses of frequent sets and condensed representations. *International Conference on Knowledge Discovery and Data Mining*, KDD'96, Portland, USA, 189-194.

Pasquier, N., Bastide, Y., Taouil, R., & Lakhal, L. (1999, January). Discovering frequent closed itemsets for association rules. *Database Theory, International Conference*, ICDT'99, Jerusalem, Israel, 398–416.

Toivonen, H. (1996). Discovery of frequent patterns in large data collections. Ph.D. Thesis, Report A-1996-5, University of Helsinki.

Tsumoto, S. (2002). Discovery of positive and negative knowledge in medical databases using rough sets. In S. & A. Shinohara (eds) *Progress in Discovery Science*, Springer, Heidelberg, 543-552.

KEY TERMS

Frequent Pattern: Pattern the support of which exceeds a user-specified threshold.

Generalized Disjunction-Free Pattern: Pattern the support of which is not determinable from the supports of its proper subsets.

Generalized Disjunctive Pattern: Pattern the support of which is determinable from the supports of its proper subsets.

Item: 1) sales product; 2) feature, attribute.

Literal: An item or negated item.

Lossless Representation of Frequent Patterns: Fraction of patterns sufficient to distinguish between frequent and infrequent patterns and to determine the supports of frequent patterns.

Pattern with Negation: Pattern containing at least one negated item.

Positive Pattern: Pattern with no negated item.

Reasoning about Patterns: Deriving supports of patterns without accessing a database.

Support of a Pattern: The number of database transactions in which the pattern occurs.

Receiver Operating Characteristic (ROC) Analysis

Nicolas Lachiche
University of Strasbourg, France

INTRODUCTION

Receiver Operating Characteristic (ROC curves) have been used for years in decision making from signals, such as radar or radiology. Basically they plot the hit rate versus the false alarm rate. They were introduced recently in data mining and machine learning to take into account different misclassification costs, or to deal with skewed class distributions. In particular they help to adapt the extracted model when the training set characteristics differ from the evaluation data. Overall they provide a convenient way to compare classifiers, but also an unexpected way to build better classifiers.

BACKGROUND

ROC analysis mainly deals with binary classifiers, models associating each entry to the positive or to the negative class. The performance of a given classifier is collected in a confusion matrix (also known as a contingency table) counting the number of training examples in each of the four cells depending on their actual classes and on their predicted classes (see Table 1).

The True Positive Rate (TPR) is the fraction of positive examples correctly classified, TP/P and the False Positive Rate (FPR) is the fraction of negative examples incorrectly classified, FP/N.

MAIN FOCUS

ROC Space

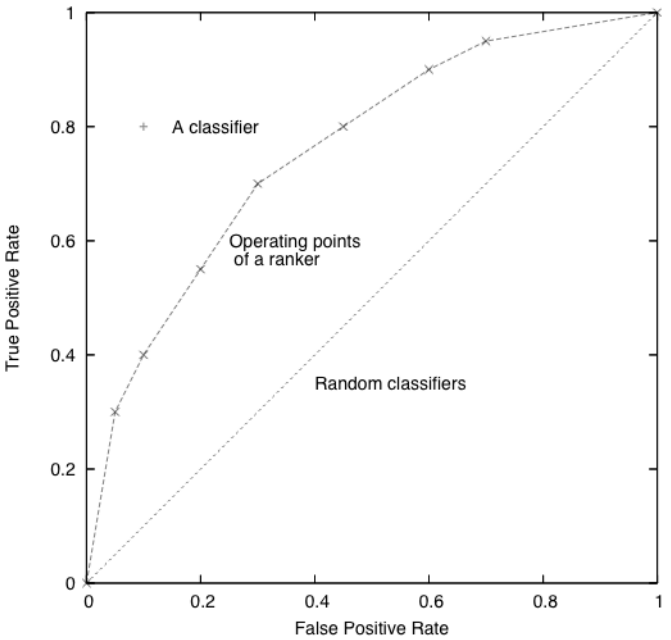
A ROC curve plots the True Positive Rate (TPR, aka recall or sensitivity) versus the False Positive Rate (FPR, equals to 1 – specificity). The performance of a binary classifier corresponds to a single point in the ROC space, cf. A classifier on Figure 1.

A ranker is a model associating a score to each entry, e.g. the naive bayesian classifier. Intuitively this score represents the "probability" that the entry is positive, even if it is not always a proper probability. Overall it provides a ranking of all examples. A threshold can be chosen and all examples whose score is above (respectively below) that threshold are predicted as positive (resp. negative). This is a called an Operating Point. It turns a ranker into a classifier. Different operating points lead to different classifiers. Therefore a ranker leads to a whole ROC curve, cf. Operating points of a ranker on Figure 1. Actually the curve is not continuous, only the operating points make sense. The number of operating points is the number of different ways of splitting the ranking of the examples. It is the number of examples if there are no tie, less otherwise. An efficient algorithm (Fawcett, 2004) to build the ROC curve is:

Table 1.

	Real Positive	Real Negative
Predicted Positive	True Positive (TP) aka Hit	False Positive (FP) aka False Alarm aka Type I Error
Predicted Negative	False Negative (FN) aka Miss aka Type II Error	True Negative (TN) aka Correct Rejection
	Total Number of Positive (P)	Total Number of Negative (N)

Figure 1. A classifier and a ranker in the ROC space



1. start in (0,0) with a score equals to the infinity
2. for each example ordered by decreasing scores
    1. if its score is different from the current score, plot a point in the current position and store its score as the current one
    2. if the example is positive, move up by 1/P, if it is negative move by 1/N

The ROC space contains several well-known landmarks:

- the top left corner corresponds to a perfect classifier, predicting correctly all positive and all negative examples,
- the bottom left corresponds to the constantly negative classifier, always predicting negative,
- the top right corner corresponds to the constantly positive classifier,
- the bottom-left to top-right diagonal denotes the performances of random classifiers, indeed any point can be obtained by a classifier randomly predicting positive a constant proportion of the entries, e.g. (0.3,0.3) corresponds to a classifier randomly predicting positive 30% of the entries.

Selection Of The Best Model According To The Operating Conditions Independently Of The Training Conditions

A ROC curve is first useful to select an operating point for a ranker or to select one among different classifiers. It should be noted that a ROC curve is insensitive to the class distribution because TPR and FPR do not depend on it. Whatever the class distribution and misclassification costs were on the training set, it is possible to select the best model according to their values on the new population to predict. Indeed once the expected number of positive P', the expected number of negative N', and the cost of misclassifying a positive (resp. negative) example C(-/+) (resp. C(+/-)) are known, it is easy to select the classifier:

- either maximizing the accuracy = P' * TPR + N' * (1 – FPR)
- or minimizing the cost = C(+/-) * FPR + C(-/+) * (1 - TPR).

Global Comparison Of Models

Classifiers are usually compared using simple measures such as the accuracy. However the accuracy depends on each operating point for a ranker. The ROC space allows to compare rankers before and independently of the final operating conditions.

- If a classifier is above and on the left-hand side of another classifier, the former dominates the latter, e.g. Classifier A dominates Classifier B on Figure 2.
- The Area Under the ROC Curve (AUC) summarizes the performances of a ranker over all possible operating points. Actually the AUC is equivalent to the Wilcoxon rank-sum test (aka Mann-Whitney U test). It measures the probability that a randomly chosen positive example will be assigned a greater rank than a randomly chosen negative example. Still one ranker can dominate another, e.g. Ranker X dominates Rankers Y and Z on Figure 2, but rankers can also be not comparable, e.g. Rankers Y and Z on Figure 2.

Improvement Of Current Models

Last but not least ROC analysis allows to build new classifiers.

- First any point between two classifiers can be reached, e.g. On Figure 3 Point C on the segment between Classifiers A and B can be reached by randomly choosing the prediction of Classifier A for 20% of the entries and the prediction of the classifier B otherwise.
- Therefore any classifier below and on the right-hand side of a segment joining two classifiers is dominated, e.g. Classifier D on Figure 3, and only the ROC convex hull is considered, cf. Figure 3.
- ROC analysis can also lead to new, better models. First, a bad (worse than random) classifier can be turned into a good classifier, by negating its predictions, e.g. Classifier A' is the negation of A on Figure 4. Flach & Wu (2005) show that a similar correction can also be applied to any classifiers B, C, D such that TPR(B) < TPR(C) < TPR(D) and FPR(B) < FPR(C) < FPR(D) and classifier C' can be built that agrees with B and D when B and D do agree, and predicts the negation of C when B and D disagree, cf. Figure 4. Those conditions are trivially satisfied if B, C and D are different operating points of a same ranker.
- If two classifiers are independent, in particular when they do not correspond to two operating points of the same ranker, Fawcett (2004) shows that their logical conjunction and disjunctions cor-

Figure 2. Comparison of models

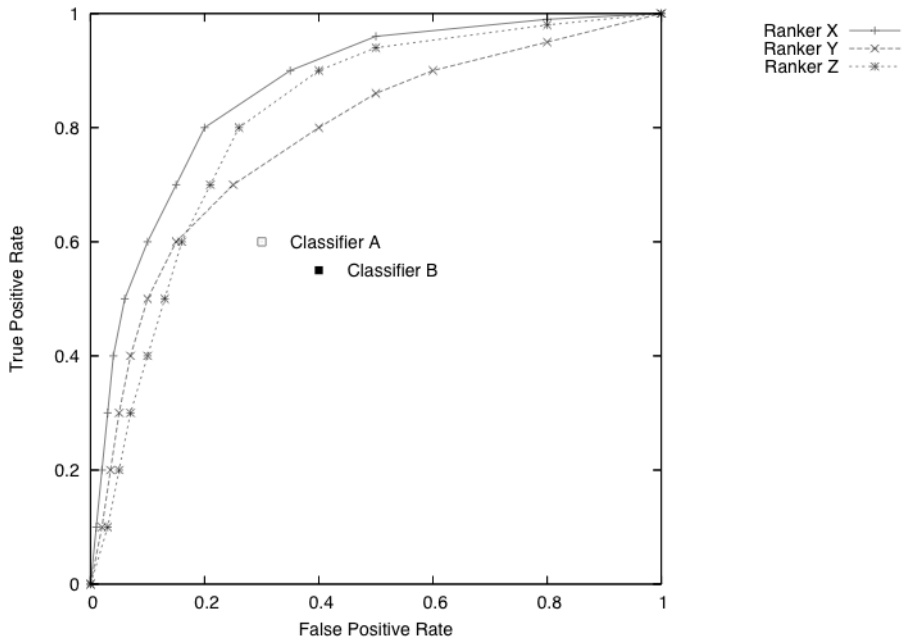


Figure 3. Interpolating classifiers and the ROC convex hull

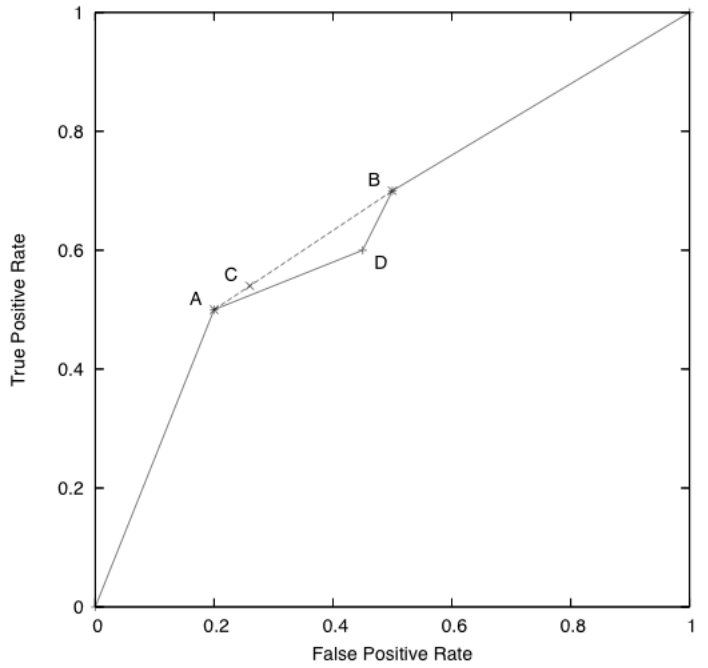


Figure 4. Improving models

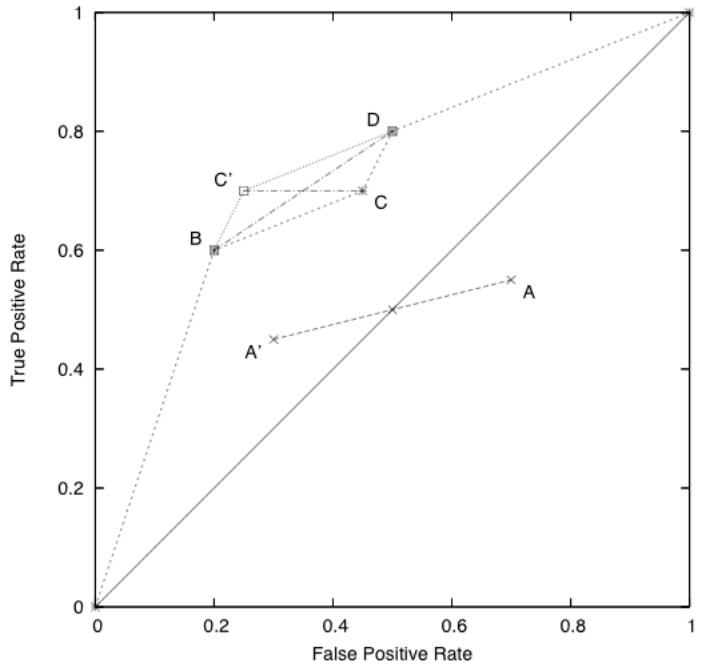
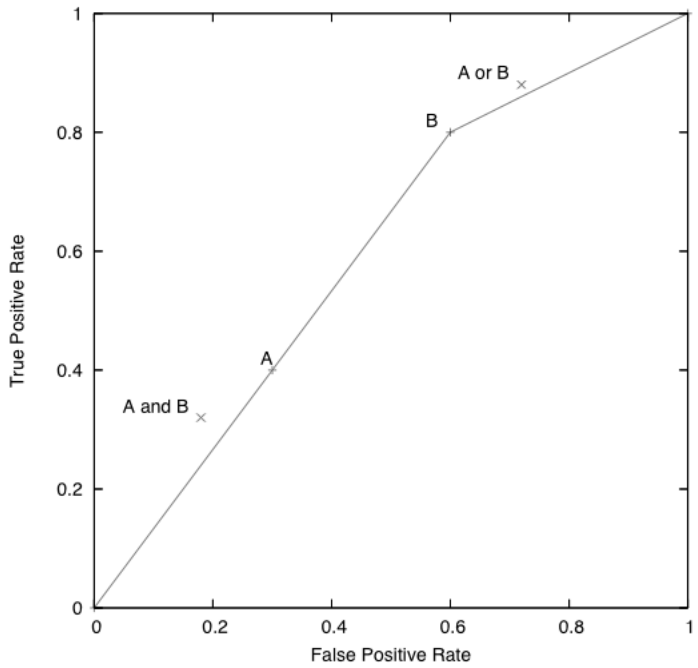


Figure 5. Logical combination of independent classifiers



respond to new classifiers whose true positive and false positive rates can be approximated by:

- $TPR(A \otimes B) = TPR(A)*TPR(B)$
- $FPR(A \otimes B) = FPR(A)*FPR(B)$
- $TPR(A \oplus B) = TPR(A) + TPR(B) - TPR(A)*TPR(B)$
- $FPR(A \oplus B) = FPR(A) + FPR(B) - FPR(A)*FPR(B)$

Those new classifiers lie outside the ROC convex hull of classifiers A and B, cf. Figure 5.

Averaging ROC Curves

First of all, a k-fold cross-validation produces k classifiers or rankers. Their results can be averaged, as their accuracies usually are, but their variance should be measured and represented too. The most frequent methods are:

- Vertical averaging: for each FPR, the average and standard deviation of the TPR on the k curves are computed,
- Threshold averaging: for each threshold, the averages and standard deviations of the TPR and FPR are computed.

Vertical averaging is practical only if the FPR can be easily tuned, and it only measures variations on the TPR dimension. Threshold averaging solves both problems. However it requires to compare the scores produced by different rankers, which may not be calibrated (Fawcett, 2004). Macskassy and Provost (2004) survey more sophisticated ways of providing confidence bands to ROC curves.

FUTURE TRENDS

ROC curve have a clear meaning since they display the hit rate versus the false alarm rate. Though alternative have been suggested that present different and useful points of view. For instance, DET (Detection Error Trade off) curves (Martin et al, 1997) plot false negative versus false positive in log scale in order to zoom on the lower left part of the graph. Indeed this part corresponds to the upper left part of the corresponding ROC curve and so helps to distinguish classifiers performances. Cost curves (Drummond & Holte, 2006) have a complementary point of view. They display the error rate in function of the proportion of positive instances, respectively the cost in function of the probability times cost.

ROC curve are primarily defined for binary classes. However lots of domains involve either multiple classes or continuous classes. Different approaches have been suggested to address multi-class problems, ranging from turning the multi-class problem into several pairwise problems (Hand & Till, 2001) to exploring a N dimensional ROC surface (Lachiche & Flach, 2003). Regression Error Curves (REC) were proposed to deal with numerical targets. Bi and Bennet (2003) prove statistical properties of the REC curve. However their usefulness is limited to graphically comparing models until now.

CONCLUSION

ROC curves have quickly become part of the standard data mining toolbox. While most people are only aware of the AUC to replace the accuracy in comparing models, ROC analysis offers other benefits: from adapting a model to different operating conditions, to building better classifiers. ROC analysis has also been used to enlight most machine learning metrics (Flach, 2003), (Fürnkranz & Flach, 2003) or to prove the equivalence between the Pool Adjacent Violators (PAV) algorithm and the ROC convex hull (Niculescu-Mizil & Fawcett, 2007). ROC analysis has still lots to offer to data mining practitioners.

REFERENCES

Bi, J., & Bennett, K. P. (2003). Regression Error Characteristic Curves. In T. Fawcett. & N. Mishra (Eds.), *Proceedings of the twentieth International Conference on Machine Learning (ICML'03)*, pp. 43-50. AAAI Press.

Drummond, C., & Holte, R. C. (2006). Cost curves: An improved method for visualizing classifier performance. *Machine Leaning*, 65:95-130. Springer.

Fawcett, T. (2004). *ROC Graphs: Notes and Practical Considerations for Researchers.* Technical report HPL-2003-4, HP Laboratories, Palo Alto, CA, USA. Available at http://www.purl.org/NET/tfawcett/papers/ROC101.pdf.

Flach, P. A. (2003). The geometry of ROC space: understanding machine learning metrics through ROC isometrics. In T. Fawcett. & N. Mishra (Eds.), *Proceedings of the twentieth International Conference on Machine Learning (ICML'03)*, pp. 194–201. AAAI Press.

Flach, P. A., & Wu, S. (2005). Repairing concavities in ROC curves. In *Proceedings of the nineteenth International Joint Conference on Artificial Intelligence (IJCAI'05)*, pp. 702-707.

Fürnkranz, J., & Flach, P. A. (2003). An analysis of rule evaluation metrics. In T. Fawcett. & N. Mishra (Eds.), *Proceedings of the twentieth International Conference on Machine Learning (ICML'03)*, pp. 202-209. AAAI Press.

Hand, D. J., & Till, R. J. (2001). A Simple Generalisation of the Area Under the ROC Curve for Multiple Class Classification Problems. *Machine Learning*, 45:171-186.

Lachiche, N., & Flach, P. A. (2003). Improving accuracy and cost of two-class and multi-class probabilistic classifiers using ROC curves. In T. Fawcett. & N. Mishra (Eds.), *Proceedings of the twentieth International Conference on Machine Learning (ICML'03)*, pp. 416–423. AAAI Press.

Macskassy, S. A., & Provost, F. J. (2004). Confidence Bands for ROC Curves: Methods and an Empirical Study. In J. Hernandez-Orallo, C. Ferri, N. Lachiche, & P. Flach (Eds.), *Proceedings of the first international workshop on ROC analysis in Artificial Intelligence (ROCAI'04)*, pp. 61-70. Available at http://www.dsic.upv.es/~flip/ROCAI2004/papers/08-macskassy-roc-bands-camera-A4.pdf

Martin, A., Doddington, G., Kamm, T., Ordowski, M., & Przybocki, M. (1997). The DET curve in assessment of detection task performance. In *Proceedings of the fifth European Conference on Speech Communication and Technology*, vol. 4, pp. 1895-1898.

Niculescu-Mizil, A., & Fawcett, T. (2007). PAV and the ROC convex hull. *Machine Learning*, 68:97-106. Springer.

KEY TERMS

Accuracy: The fraction of examples correctly classified, i.e. the number of positive examples classified as positive plus the number of negative examples

classified as negative divided by the total number of examples.

AUC: The Area Under the ROC Curve.

Classifier: A model associating a class to each entry, and more generally the learning system that extracts that model from the training data.

Cost-Sensitive: Taking into account different costs associated to different misclassifications, for instance the loss associated to granting a credit to a faulty customer compared to rejecting a good customer.

Operating Point: One point of a ROC curve, corresponding to one threshold of a ranker, i.e. to one classifier. The threshold can be expressed in terms of the corresponding respective prior probabilities of positive and negative.

Ranker: A model associating a rank to each entry, and more generally the learning system that extracts that model from the training data.

ROC Curve: The Receiver Operating Characteristic curve plots the hit rate in function of the false alarm rate, and was first used with Radars.

ROC Convex Hull: The convex envelop of the ROC curve.

Reflecting Reporting Problems and Data Warehousing

Juha Kontio
Turku University of Applied Sciences, Finland

INTRODUCTION

Reporting is one of the basic processes in all organizations. Reports should offer relevant information for guiding the decision-making. Reporting provides information for planning and on the other hand it provides information for analyzing the correctness of the decisions made at the beginning of the processes. Reporting is based on the data the operational information systems contain. Reports can be produced directly from these operational databases, but an operational database is not organized in a way that naturally supports analysis. An alternative way is to organize the data in such a way that supports analysis easily. Typically this leads to the introduction of a data warehouse.

In summer 2002 a multiple case study research was launched in six Finnish organizations. The research studied the databases of these organizations and identified the trends in database exploitation. One of the main ideas was to study the diffusion of database innovations. In practice this meant that the present database architecture was described and the future plans and present problems were identified. The data was mainly collected with semi-structured interviews and altogether 54 interviews were arranged.

The research processed data of 44 different information systems. Most (40 %) of the analyzed information systems were online transaction processing systems like order-entry systems. Second biggest category (30 %) was information systems relating to decision support and reporting. Only one pilot data warehouse was among these, but on the other hand customized reporting systems was used for example in SOK, SSP and OPTI. Reporting was anyway commonly recognized as an area where interviewees were not satisfied and improvements were hoped.

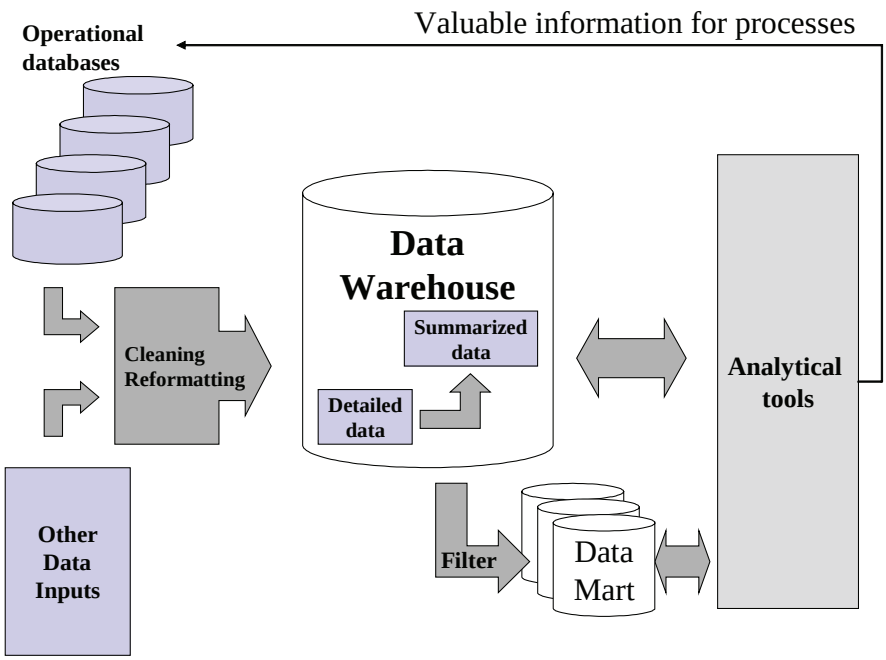
Turku University of Applied Sciences is one of the largest of its kind in Finland with almost 9000 students and 33 Degree Programs. Our University is organized in six units of education that promote multidisciplinary learning. In autumn 2005 an enterprise resource planning system was introduced at Turku University of Applied Sciences. At the heart of this information system is a data warehouse collecting necessary information from the operational databases.

This paper concentrates on briefly describing the identified problems in reporting in the earlier research and how a data warehouse might help overcoming these problems (a more thorough description is provided at (Kontio 2005)). These ideas are benchmarked with usage experiences of the data warehouse based ERP at Turku University of Applied Sciences resulting to some generalizations and confirmation.

BACKGROUND

The term data warehouse was first introduced as a subject-oriented, integrated, non-volatile, and time variant collection of data in support of management's decisions (Inmon 1992). A simpler definition says that a data warehouse is a store of enterprise data that is designed to facilitate management decision-making (Kroenke 2004). A data warehouse differs from traditional databases in many points. The structure of a data warehouse is different than traditional databases and different functionalities are required (Elmasri and Navathe 2000). The aim is to integrate all corporate information into one repository where the information is easily accessed, queried, analyzed and used as basis for the reports (Begg and Connolly 2002). A data warehouse provides decision support to organizations with the help of analytical databases and On Line Analytical Processing (OLAP) tools (Gorla 2003). A data warehouse (see Figure 1.) receives data from the operational databases on regular basis and new data is added to the existing data. This process is called an ETL-process letters standing for extract, transform and load (Simitsis 2005). A data warehouse contains both detailed aggregated data and also summarized data to

Figure 1. Data Warehousing



speed up the queries. A data warehouse is typically organized in smaller units called data marts that support the specific analysis needs of a department or a business unit (Bonifati, Cattaneo et al. 2001).

In the case organizations the idea of the data warehouse had been discussed but so far no data warehouses existed although in one case a data warehouse pilot was in use. The main reason for these discussions was that the reporting and the analyzing possibilities are not serving the organizations very well. The research actually identified many problems in reporting. The reporting and analyzing problems played also a role in starting the ERP design at our University.

The case organizations of the 2002 study introduced several problems in reporting and analyzing (Table 1).

These cases proposed alternative solutions to overcome the presented problems in reporting and analyzing. In case A centralization of the data has been discussed to overcome the challenges. In case B the organization has discussed a data warehouse solution for three

Table 1. Identified problems in reporting and analyzing.

Organization	Problems in reporting and analyzing
A	The data is distributed in numerous information systems and building a comprehensive view of the data is difficult. Financial reporting gives conflicting results, because data is not harmonized and processed similarly.
B	The major information system is somehow used inconsistently leading to data that is not consistent. The reporting system has capacity limitations and users are incapable to define own reports. Analyzing customer data is also difficult, because the collection of relevant data is very hard.
C	The case organization gathers and analyzes a large amount of data, but no special data management tools were not in use. The analysis task is a very burdensome task.
D	Reporting was not identified as a major problem at the moment, but a DW might offer extra value in data analysis.
E	A DW pilot existed, but the distribution and the format of the reports should be solved. Due to some compatibility problems the usage of the system is not widely spread.
F	Reports are generated directly from operational databases that are not designed for reporting purposes.

reasons: a) to get rid of the capacity problems, b) to provide a user friendlier system where definition of reports is easier and c) to gain more versatile analysis possibilities. In case F a data warehouse solution was justified with the possibility to define customized reports. In addition, the ability to serve customers better was named as a rationale.

Before 2005, we used balanced scorecards to define our goals. We had separate Excel sheets for defining BSC numeric goals. Unfortunately, these goals remained partly irrelevant and distant from the operational level. Reliable analysis of our success was difficult to achieve, because exact data was not easily available. This influenced our reporting as well. Our reporting and analysis was unformulated and random and it seldom had direct effects on our operation. In addition, there were certain mandatory reports that required a plenty of manual work. We had to give regular reports to the Ministry of Education, City of Turku and some other public organizations as well.

At 2005, an ERP system was introduced for management and resource planning in our institute. The introduction of this system was an effort to link operational and strategic management. Resource planning is here understood as setting goals, planning actions for the goals, resourcing the actions and follow-up and analysis of the results. A central part of our ERP system is a data warehouse where operational data is extracted, transformed and loaded. This data is used for analysis, action planning and reporting.

MAIN THRUST OF THE CHAPTER

Before introducing a data warehouse the organization has to do a lot of careful design. A requirement analysis is needed to successfully design and represent the desired information needs in the data warehouse (Mazon, Trujillo et al. 2007). Designing a data warehouse requires quite different techniques than the design of an operational database (Golfarelli and Rizzi 1998). Modeling a data model for a data warehouse is seen as one of the most critical phases in the development process of a data warehouse (Bonifati, Cattaneo et al. 2001). This data modeling has specific features that distinguish it from normal data modeling (Busborg, Christiansen et al. 1999). At the beginning the content of the operational information systems, the interconnections between them and the equivalent entities should be understood (Blackwood 2000). It is crucial to obtain a consistent reconciliation of data sources and information needs (Mazon, Trujillo et al. 2007). In practice this means studying the data models of the operational databases and developing an integrated schema to enhance data interoperability (Bonifati, Cattaneo et al. 2001).

The data modeling of a data warehouse is called dimensionality modeling (DM) (Golfarelli and Rizzi 1998; Begg and Connolly 2002). Dimensional models were developed to support analytical tasks (Loukas and Spencer 1999). Dimensionality modeling concentrates on facts and the properties of the facts and dimensions connected to facts (Busborg, Christiansen et al. 1999). Facts are numeric and quantitative data of the business and dimensions describe different dimensions of the business (Bonifati, Cattaneo et al. 2001). In other words, facts are used to memorize quantitatively business situations and dimension are used to analyze these situations (Schneider 2007). Fact tables contain all business events to be analyzed and dimension tables define how to analyze fact information (Loukas and Spencer 1999). The result of the dimensionality modeling is typically presented in a star model or in a snowflake model (Begg and Connolly 2002). multi dimensional schema (MDS) is a more generic term that is used to collectively refer to both schemas (Martyn 2004). When a star model is used the fact-tables are normalized, but dimension-tables are not. When dimension-tables are normalized too, the star-model turns into snowflake-model. (Bonifati, Cattaneo et al. 2001)

Ideally an information system like a data warehouse should be correct, fast and friendly (Martyn 2004). Correctness is especially important in data warehouses to ensure that decisions are based on correct information. In principal data warehouse is used for strategic decisions, hence the quality of data warehouse is crucial (Serrano, Trujillo et al. 2007). However, it is estimated that 30 to 50 percent of information in a typical database is missing or incorrect (Blackwood 2000). This emphasizes the need to pay attention to the quality of the source data in the operational databases (Finnegan and Sammon 2000). A problem with MDS is that when the business environment changes the evolution of MD schemas is not as manageable as normalized schemas (Martyn 2004). It is also said that any architecture not based on third normalized form can cause the failure of a data warehouse project (Gardner 1998). On the other hand however a dimensional model provides a

better solution for a decision support application than a pure normalized relational model (Loukas and Spencer 1999). All the above is actually related to efficiency since a large amount of data is processed during analysis. Typically this is a question about needed joins in the database level. Usually a star schema is the most efficient design for a data warehouse since the denormalized tables require fewer joins (Martyn 2004). However recent developments in storage technology, access methods (like bitmap indexes) and query optimization are indicating that the performance with the third normalized form should be tested before moving to multi dimensional schemas (Martyn 2004). From this 3NF schema a natural step toward MDS is to use denormalization and this will support both efficiency and also flexibility issues (Finnegan and Sammon 2000). Still, there is a possibility to define necessary SQL views on top of the 3NF schema without denormalization (Martyn 2004). Finally, during the design, the issues of physical and logical design should be separated: physical is about performance and logical is about understandability (Kimball 2001).

The OLAP tools access data warehouse for complex data analysis and decision support activities (Kambayashi, Kumar et al. 2004). OLAP tools are based on MDS views. These tools include typically assessing the effectiveness of a marketing campaign, product sales forecasting and capacity planning. The architecture of the underlying database of the data warehouse categorizes the different analysis tools. (Begg and Connolly 2002) Depending on the schema type terms relational OLAP (ROLAP), multidimensional OLAP (MOLAP) and Hybrid OLAP (HOLAP) are used (Kroenke 2004). ROLAP is a preferable choice when a) the information needs change frequently, b) the information should be as current as possible and c) the users are sophisticated computer users (Gorla 2003). The main differences between ROLAP and MOLAP are in currency of data and in data storage processing capacity. MOLAP populates an own structure of the original data when it is loaded from the operational databases (Dodds, Hasan et al. 2000). In MOLAP the data is stored in a special-purpose MDS (Begg and Connolly 2002). ROLAP analyzes the original data and the users can drill down to the unit data level (Dodds, Hasan et al. 2000). ROLAP uses a meta-data layer to avoid the creation of a MDS. ROLAP typically utilize the SQL extensions like CUBE and ROLLUP in Oracle DBMS (Begg and Connolly 2002).

FUTURE TRENDS

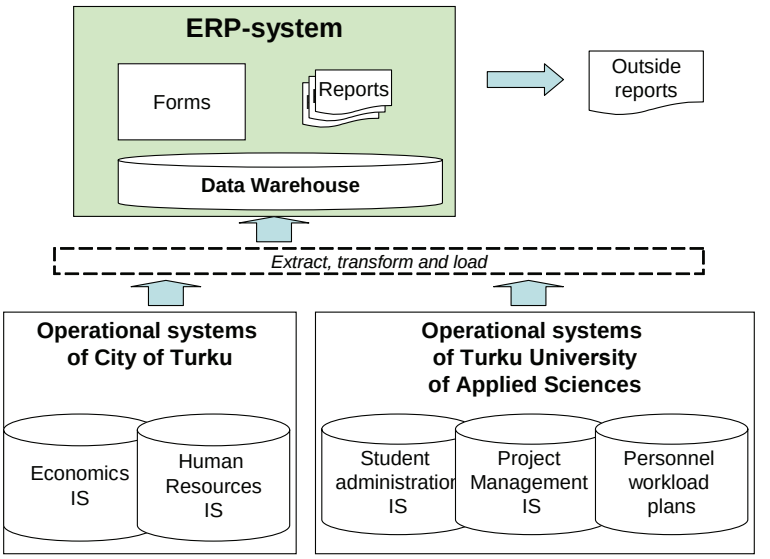
Several future steps were identified in the 2002 case study (Table 2). The presented cases reflect probably quite well the overall situation in different organizations and we might give predictions what is happening in the future. The cases show that organizations have clear problems in analyzing operational data. They have developed own applications to ease the reporting, but still they are living with inadequate solutions. data warehousing has been identified as a possible solution for the reporting problems and the future will show whether data warehouses diffuse in the organizations.

In our university our main operational systems are integrated into the data warehouse of our ERP system (Figure 2). Our future with the ERP system and the data warehouse concentrates on learning the new way of working and learning to utilize the new possibilities in a reasonable manner. There are for example differ-

Table 2. Future of reporting and analyzing in the cases.

Organization	Future steps
A	A comprehensive view of the corporation data needs to be built i.e. an enterprise data model.
B	A major concern is correctness of the data and the way users work with the information systems. The consistency of the data has to be improved before starting a data warehouse project.
C	At first a suitable tool/system is needed for managing the operational data. This provides possibilities to truly benefit from data analysis.
D	A data warehouse could define standard formats for the reports and it could also solve the report distribution problems.
E	A data warehouse could solve slowness problems in reporting. Introduction necessary OLAP tools will improve the situation even further.
F	A data warehouse might offer extra value in data analysis when it is introduced - if at all.

Figure 2. ERP and DW at Turku University of Applied Sciences.



ent levels of planning and analyzing in the system and the lowest levels are learning while higher levels are fully utilizing the system and the data offered from the data warehouse. Furthermore, in the future additional features like more detailed reporting possibilities will be introduced.

CONCLUSION

The case organization recognized the possibilities of a data warehouse to solve the problems in reporting. In our organization we are much longer in our implementation process than the cases were at the time of the research. Now we will compare the findings of the 2002 case study and our experiences with a real life data warehouse implementation.

The data warehouse initiatives of the case organizations based on business requirements, which is a good starting point because to success a data warehouse should be a business-driven initiative, but in partnership with the IT department (Gardner 1998; Finnegan and Sammon 2000). However the first step should be the analysis of the real needs of a data warehouse. Our case also started from business needs: to connect strategic and operational management. This required an environment were necessary data is available for analysis and reporting purposes. The success of the system confirms the essential starting point: business needs.

To really support reporting the operational data should be organized more like in a data warehouse. In practice this means studying and analyzing the existing operational databases. As a result, a data model describing the necessary elements of the data warehouse should be achieved. A suggestion is that first an enterprise level data model is produced to describe all the data and the interconnections of it. This enterprise level data model will be the basis for the data warehouse design. As the theory suggested at the beginning a normalized data model with SQL views should be produced and tested. This can further be denormalized into snowflake or into star model when performance requirements are not met with the normalized schema. Our case pretty much confirms also these ideas. At the beginning a lot of work was done with defining the data, defining the entities and their properties. A database of our data and their definitions was created. A data model was also designed describing the data that is needed in the future data warehouse. Finally, a data warehouse schema was defined with fact and dimension tables.

Producing a data model for the data warehouse is only one part. In addition special attention should be taken in designing data extraction from operational databases to the data warehouse. The enterprise level data model helps understanding the data in these databases and thus eases the development of data extraction solutions. When the data warehouse is in use there should be automated tools that speed up the loading the operational data into the data warehouse (Finnegan and Sammon

2000). Our case confirms that one of the biggest challenges is the ETL-process. It also emphasizes careful planning and design which operational information systems are connected and which are not.

Before loading data into the data warehouse the organizations should also analyze the correctness and the quality of the data in the operational databases. In our case a series of careful test were done and the correctness of the data has been checked over an over. Our case also highlights the importance of the correct use of the operational systems. The data in operational systems must be clean and correct. The more variants and problems there are with the data in operational databases the more difficult it becomes to design the ETL-process successfully. It is essential that the data received from data warehouse is reliable. At the beginning we had some problems with this data. This caused additional problems and extra work thus resulting in just opposite consequences than a data warehouse was supposed to result. Nowadays most of these problems of dirty data are solved. As our experiences show this phase is very important. The technical ETL process might be free of errors, but when the operational data is dirty you cannot be delighted.

Finally, as the paper showed, a data warehouse is a relevant alternative for solving the problems in reporting that the case organizations are currently working with. After the implementation of a Data Warehouse organizations must ensure that the possible users of the system are educated to fully take advantage of the new possibilities that the DW offers. During our implementation process many education sessions had been organized. The system was built and implemented in phases and the users were trained at the same pace. This made the adoption of the system easier. Quite often the training sessions concentrated on the technical features and how to do different single things. The overall management process and the relationship of these different single things were difficult to figure out at the beginning. Still, we see that the implementation of the ERP system has been successful and the training of the user has been sufficient. However, organizations should remember that there is no quick jump to data warehouse exploitation rather all steps takes time and patience, but as this study confirms it is all worth of it.

REFERENCES

Begg, C. and T. Connolly (2002). *Database Systems: A Practical Guide to Design, Implementation, and Management*, Addison-Wesley.

Blackwood, P. (2000). Eleven steps to success in data warehousing. *Business Journal 14*(44), 26 - 27.

Bonifati, A., F. Cattaneo, et al. (2001). Designing data marts for data warehouses. *ACM Transactions on Software Engineering and Methodology 10*(4), 452 - 483.

Busborg, F., J. G. B. Christiansen, et al. (1999). starER: a conceptual model for data warehouse design. *ACM international workshop on Data warehousing and OLAP*, Kansas City, Missouri, ACM Press.

Dodds, D., H. Hasan, et al. (2000). Approaches to the development of multi-dimensional databases: lessons from four case studies. *ACM SIGMIS Database 3*1(3): 10 - 23.

Elmasri, R. and S. B. Navathe (2000). *Fundamentals of Database Systems.* Reading, Massachussetts, Addison-Wesley.

Finnegan, P. and D. Sammon (2000). The ten commandments of Data Warehousing. *ACM SIGMIS Database 31*(4), 82 - 91.

Gardner, S. R. (1998). Building the data warehouse. *Communications of the ACM 41*(9): 52 - 60.

Golfarelli, M. and S. Rizzi (1998). A methodological framework for data warehouse design. *ACM international Workshop on Data warehousing and OLAP*, Washington, D.C., United States, ACM Press.

Gorla, N. (2003). Features to consider in a data warehousing system. *Communications of the ACM 46*(11), 111 - 115.

Inmon, W. H. (1992). *Building the Data Warehouse*. New York, NY, USA, John Wiley & Sons, Inc.

Kambayashi, Y., V. Kumar, et al. (2004). *Recent Advances and Research Problems in Data Warehousing*. Lecture Notes in Computer Science 1552: 81 - 92.

Kimball, R. (2001). *A trio of interesting snowflakes. Intelligent Enterprise 30*(4) - 32.

Kontio, J. (2005). Data warehousing solutions for reporting problems. In J. Wang (Ed.) *Encyclopedia of Data warehousing and Mining*. Hershey, PA: Idea Group Publishing.

Kroenke, D. M. (2004). *Database Processing: Fundamentals, Design and Implementation*. Upper Saddle River, New Jersey, Pearson Prentice Hall.

Loukas, T. and T. Spencer (1999). From star to snowflake to erd: comparing data warehouse design approaches. *Enterprise Systems* (10/99)

Martyn, T. (2004). Reconsidering multi-dimensional schemas. *ACM SIGMOD Record 33*(1), 83-88.

Mazon, J.-N., J. Trujillo, et al. (2007). Reconciling requirement-driven data warehouses with data sources via multidimensional normal forms. *Data & Knowledge Engineering 63*(3), 725-751.

Schneider, M. (2007). A general model for the design of data warehouses. *International Journal of Production Economics*

Serrano, M., J. Trujillo, et al. (2007). Metrics for data warehouse conceptual models understandability. *Information and Software Technology 4*9(8), 851-870.

Simitsis, A. (2005). *Data warehouse design 2: Mapping a conceptual logical models for ETL processes*. The 8th ACM international workshop on Data warehousing and OLAP DOLAP '05, ACM Press.

KEY TERMS

Data Extraction: A process where data is transferred from the operational databases to the data warehouse. Also term ETL (extract-transform-load) -process is used to give a more realistic picture of the process that needs to be done to move data from operational databases to a data warehouse.

Dimensionality Modeling: A logical design technique that aims to present data in a standard, intuitive form that allows for high-performance access.

Normalization/Denormalization: A technique for producing a set of relations with desirable properties, given the data requirements of an enterprise. Denormalization is a step backward in normalization process for example to improve performance.

OLAP: The dynamic synthesis, analysis, and consolidation of large volumes of multi-dimensional data.

Snowflake Model: A variant of the star schema where dimensions tables do not contain denormalized data.

Star Model: A logical structure that has a fact table containing factual data in the center, surrounded by dimension tables containing reference data.

Robust Face Recognition for Data Mining

Brian C. Lovell
The University of Queensland, Australia

Shaokang Chen
NICTA, Australia

Ting Shan
NICTA, Australia

INTRODUCTION

While the technology for mining text documents in large databases could be said to be relatively mature, the same cannot be said for mining other important data types such as speech, music, images and video. Multimedia data mining attracts considerable attention from researchers, but multimedia data mining is still at the experimental stage (Hsu, Lee & Zhang, 2002). Nowadays, the most effective way to search multimedia archives is to search the metadata of the archive, which are normally labeled manually by humans. This is already uneconomic or, in an increasing number of application areas, quite impossible because these data are being collected much faster than any group of humans could meaningfully label them — and the pace is accelerating, forming a veritable explosion of non-text data. Some driver applications are emerging from heightened security demands in the 21st century, postproduction of digital interactive television, and the recent deployment of a planetary sensor network overlaid on the internet backbone.

BACKGROUND

Although they say a picture is worth a thousand words, computer scientists know that the ratio of information contained in images compared to text documents is often much greater than this. Providing text labels for image data is problematic because appropriate labeling is very dependent on the typical queries users will wish to perform, and the queries are difficult to anticipate at the time of labeling. For example, a simple image of a red ball would be best labeled as sports equipment, a toy, a red object, a round object, or even a sphere, depending on the nature of the query. Difficulties with text metadata have led researchers to concentrate on techniques from the fields of Pattern Recognition and Computer Vision that work on the image content itself. Although pattern recognition, computer vision, and image data mining are quite different fields, they share a large number of common functions (Hsu, Lee & Zhang, 2002).

An interesting commercial application of pattern recognition is a system to semi-automatically annotate video streams to provide content for digital interactive television. A similar idea was behind the MIT MediaLab Hypersoap project (The Hypersoap Project, 2007; Agamanolis & Bove, 1997). In this system, users touch images of objects and people on a television screen to bring up information and advertising material related to the object. For example, a user might select a famous actor and then a page would appear describing the actor, films in which they have appeared, and the viewer might be offered the opportunity to purchase copies of their other films. In the case of Hypersoap, the metadata for the video was created manually. Automatic face recognition and tracking would greatly simplify the task of labeling video in post-production — the major cost component of producing such interactive video.

With the rapid development of computer networks, some web-based image mining applications have emerged. SIMBA (Siggelkow, Schael, & Burkhardt, 2001) is a content-based image retrieval system performing queries based on image appearance from an image database with about 2500 images. RIYA (RIYA Visual Search) is a visual search engine that tries to search content relevant images from the context input. In 2007, Google added face detection to its image search engine (Google Face Search). For example, the URL *http://images.google.com/images?q=bush&imgtype=face* will return faces associated with the name "Bush" including many images of recent US presidents. While

the application appears to work well, it does not actually identify the face images. Instead it relies on the associated text metadata to determine identity.

None of the above systems support the input of face images as a query to retrieve similar images of the same person. A robust face recognition method is needed for such kind of systems. Now we will focus on the crucial technology underpinning such a data mining service—automatically recognizing faces in image and video databases.

MAIN FOCUS

Challenges for Face Recognition

Robust face recognition is a challenging goal because differences between images of the same face (intraclass variation) due to nuisance variations in lighting conditions, view point, pose, age, health, and facial expression are often much greater than those between different faces (interclass variation) (Adinj, Moses & Ulman, 1997, Zhao, Chellappa, Philips, & Rosenfeld, 2003) An ideal face recognition system should recognize new images of a known face and be insensitive to nuisance variations in image acquisition. Most systems work well only with images taken under constrained or laboratory conditions where lighting, pose, and camera parameters are strictly controlled. This requirement is much too strict to be useful in many data mining situations when only few sample images are available such as in recognizing people from surveillance videos or searching historic film archives. The following is a list of three key problems existing for current face recognition technology:

- Overall accuracy, particularly on large databases
- Sensitivity to changes in lighting, expression, camera angle, pose
- Computational load of searches

Illumination Invariant Face Recognition

Recent research has been focused on diminishing the impact of nuisance factors on face recognition. Two main approaches have been proposed for illumination invariant recognition. The first is to represent images with features that are less sensitive to illumination change (Gao & Leung, 2002) such as using the edge maps of an image. However, the locality problem of edge representation due to small rotation or location errors will degenerate the performance greatly. Yilmaz and Gokmen (2000) overcome the locality problem by using hills to spread the edge profile. These methods assume that features do not change dramatically with variable lighting conditions. Yet this is patently false as edge features generated from shadows may have a significant impact on recognition. Experiments done by Adinj et al. (Adinj, Moses, & Ulman, 1997) show that even when using the best illumination insensitive features for image presentation, the classification error is more than 20%. The second main approach is to construct a low dimensional linear subspace for the images of faces taken under different lighting conditions. This method is based on the assumption that images of a convex Lambertian object under variable illumination form a convex cone in the space of all possible images (Belhumeur & Kriegman, 1998). Ignoring the effect of shadows, this subspace has three dimensions (Zhao & Yang 1999). To account for attached shadows 5 to 9 dimensions are needed (Basri & Jacobs, 2003; Georghiades, Belhumeur, & Kriegman, 2001). All these methods assume that the surface of human face is Lambertian reflective and convex, and thus cannot describe cast shadows. Furthermore, such systems need several images of the same face taken under different lighting source directions to construct a model of a given face — in data mining applications it is often impossible to obtain the required number of images.

Expression Invariant Face Recognition

As for expression invariant face recognition, this is still an open problem for machine recognition and it is also quite a difficult task for humans. The approach adopted in the work of Black, Fleet, and Yacoob (2000) is to morph images to the same expression as the one used for training. A problem is that not all images can be morphed correctly. For example an image with closed eyes cannot be morphed to a standard image because of the lack of texture inside the eyes. Liu, Chen, and Kumar (2003) proposed using optical flow for face recognition with facial expression variations. However, it is hard to learn the motion within the feature space to determine the expression changes, since the way one person expresses a certain emotion is normally

somewhat different from others. Martinez proposed a weighing method to deal with facial expressions (Martinez, 2002). An image is divided into several local areas and those that are less sensitive to expression change are chosen and weighed independently. But features that are insensitive to expression change may be sensitive to illumination variations. These methods also suffer from the need to have large numbers of example images for training.

Mathematical Basis for Face Recognition Technologies

Most face recognition systems are based on one of the following methods:

1. Direct Measurement of Facial Features
2. Principal Components Analysis or "Eigenfaces" (Turk & Pentland, 1991)
3. Fisher Linear Discriminant Function (Liu & Wechsler, 1998)

Early forms of face recognition were based on Method 1 with direct measurement of features such as width of the nose, spacing between the eyes, etc. These measurements were frequently performed by hand using calipers. Many modern systems are based on either of Methods 2 or 3 which are better suited to computer automation. Here we briefly describe the principles behind one of the most popular methods — Principal Components Analysis (PCA), also known as "eigenfaces," as originally popularized by Turk and Pentland (1991).

Principal Components Analysis (PCA)

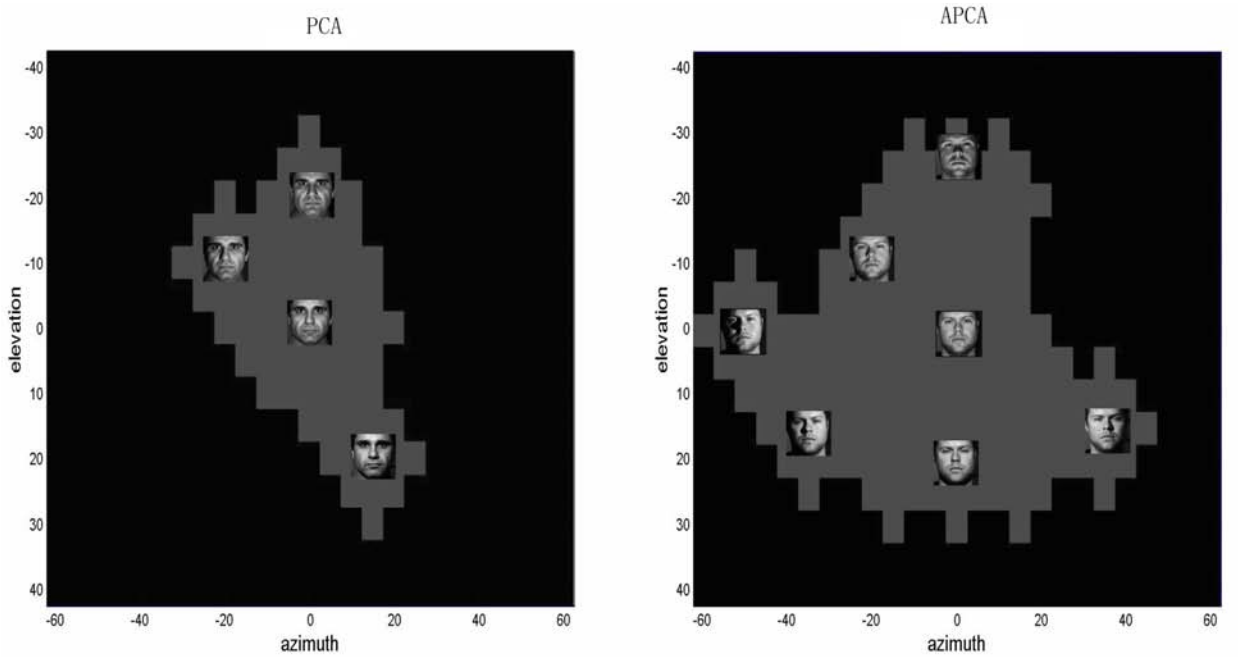
PCA is a second-order method for finding a linear representation of faces using only the covariance of the data. It determines the set of orthogonal components (feature vectors) which minimizes the reconstruction error for a given number of feature vectors. It is often used to define a face subspace with reduced dimensionality which contains the greatest covariance and yields good generalization capacity. Eigen-decomposition is normally applied to the covariance matrix C to generate eigenvectors, which are frequently called the eigenfaces and are shown as images in Figure 1. Generally, we select a small subset of eigenfaces to define a facespace which yields highest recognition performance on unseen examples of faces. For good recognition performance the required number of eigenfaces, m, is typically chosen to be of the order of 6 to 10. Nevertheless, PCA is optimized for minimizing mean square error and does not consider the classification of samples. Thus, features abstracted by PCA are preferred for representing face images, but are not the best choice for classification. Indeed some principal components may be selected that are greatly affected by within-class variation. Specifically, PCA does not provide satisfactory face recognition under different lighting conditions.

Robust PCA Recognition

The authors developed Adaptive Principal Component Analysis (APCA) to improve the robustness of PCA to nuisance factors such as lighting and expression (Chen & Lovell, 2003 and 2004). In the APCA method, we first apply PCA. Then rotate and warp the facespace

Figure 1. Typical set of eigenfaces as used for face recognition. Leftmost image is the average face.

Figure 2. Contours of 95% recognition performance for the original PCA and the proposed APCA method against lighting elevation and azimuth from Chen & Lovell (2004)



by whitening and filtering the eigenfaces according to overall covariance, between-class, and within-class covariance to find an improved set of eigenfeatures. Figure 2 shows the large improvement in robustness to lighting angle. The proposed APCA method allows us to recognize faces with high confidence even if they are half in shadow using only one gallery image. Figure 3 shows significant recognition performance gains over standard PCA when both changes in lighting and expression are present.

Pose Compensation

In 2006, the authors extended APCA to deal with face recognition under changes in pose (Shan, Lovell & Chen 2006; Sanderson, Shan & Lovell 2007). An Active Appearance Model (AAM) (Cootes, Edwards & Taylor 2001) is trained from images to learn the face shape and texture changes from different view points. A rotation model was then built to estimate pose orientation. When the pose angle is estimated, a synthesized frontal view image can be reconstructed from the model which can be used for face recognition. Figures 3 shows the recognition accuracy of APCA and PCA on face images with differing image capture conditions and Figure 4 shows reduced sensitivity to pose using the synthesized frontal face images.

After applying pose compensation to construct synthetic images, both APCA and PCA perform up to 5 times better. The method yields quite acceptable recognition rates for head rotations up to 25 degrees.

Critical Issues for Face Recognition Technology

Despite the huge number of potential applications for reliable face recognition, the need for such search capabilities in multimedia data mining, and the great strides made in recent decades, there is still much work to do before these applications become routine. Table 1 shows a summary of the critical issues for face recognition technologies.

FUTURE TRENDS

Face recognition and other biometric technologies are coming of age due to the need to address heightened security concerns in the 21st century. Privacy concerns that have hindered public acceptance of these technologies in the past are now yielding to society's need for increased security while maintaining a free society. Apart from the demands from the security sector, there

Figure 3. Recognition rates for APCA and PCA versus number of eigenfaces with variations in lighting and expression from Chen and Lovell (2003)

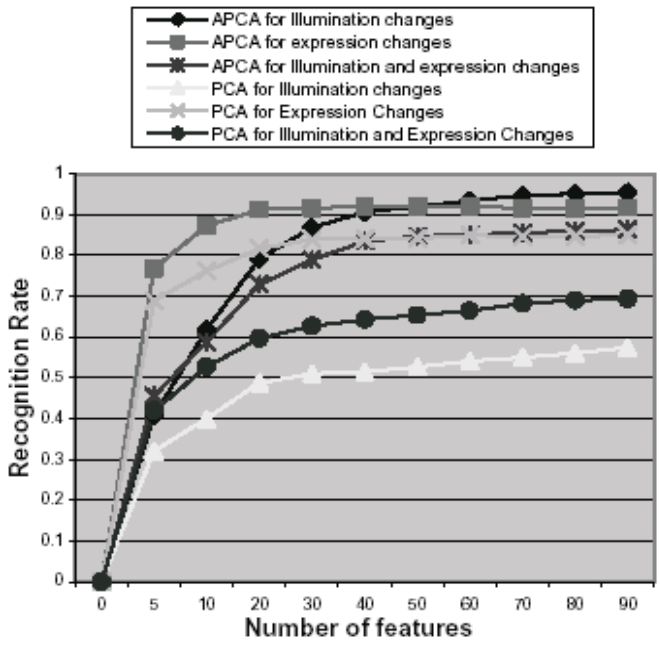
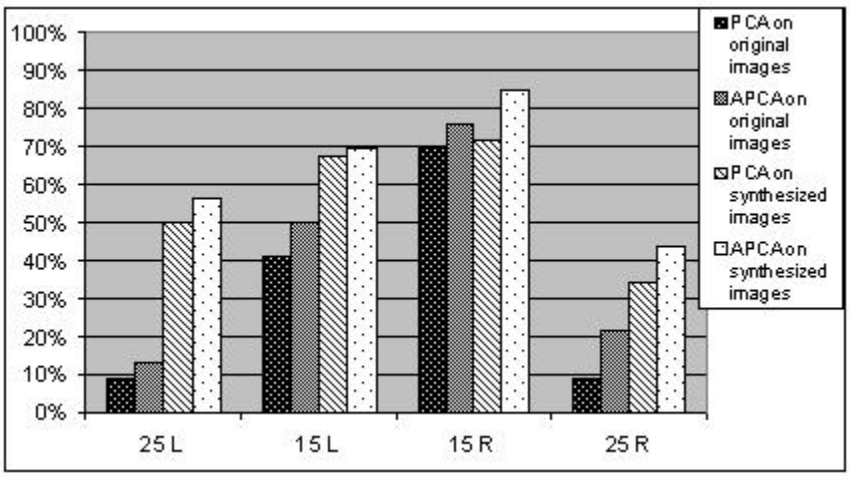


Figure 4. Recognition rate for PCA and APCA on non-frontal face images and on synthesized images from Shan, Lovell and Chen (2006)



are many applications for the technology in other areas of data mining. The performance and robustness of systems will increase significantly as more researcher effort is brought to bear. In recent real-time systems there is much interest in 3D reconstruction of the head from multiple camera angles, but in data mining the focus must remain on reliable recognition from single photos.

CONCLUSION

It has been argued that by the end of the 20th century computers were very capable of handling text and numbers and that in the 21st century computers will have to be able to cope with raw data such as images and speech with much the same facility. The explosion of multimedia data on the internet and the conversion of all information to digital formats (music, speech,

Table 1. A summary of critical issues of face recognition technologies

Privacy Concerns	It is clear that personal privacy may be reduced with the widespread adoption of face recognition technology. However, since2001 and the 911 attack, concerns about privacy have taken a back seat to concerns about personal security. Governments are under intense pressure to introduce stronger security measures. Unfortunately governments' current need for robust biometric technology does nothing to improve performance in the short term and may actually damage uptake in the medium term due to unrealistic expectations.
Computational Efficiency	Face recognition can be computationally very intensive for large databases. This is a serious impediment for multimedia data mining.
Accuracy on Large Databases	Studies indicate that recognition error rates of the order of 10% are the best that can be obtained on large databases. This error rate sounds rather high, but trained humans do no better and are much slower at searching (Face Recognition Vendor Test 2006)
Sensitivity to Illumination and Other Changes	Changes in lighting, camera angle, and facial expression can greatly affect recognition performance.
Inability to Cope with Multiple Head Poses	Very few systems can cope with non-frontal views of the face. Some researchers propose 3D recognition systems using stereo cameras for real-time applications, but these are not suitable for data mining.
Ability to Scale	While a laboratory system may work quite well on 20 or 30 faces, it is not clear that these systems will scale to huge face databases as required for many security applications such as detecting faces of known criminals in a crowd or the person locator service on the planetary sensor web.

television) is driving the demand for advanced multimedia search capabilities, but the pattern recognition technology is mostly unreliable and slow. Yet, the emergence of handheld computers with built-in speech and handwriting recognition ability, however primitive, is a sign of the changing times. The challenge for researchers is to produce pattern recognition algorithms, such as face recognition, reliable and fast enough for deployment on data spaces of a planetary scale.

REFERENCES

Adinj, Y., Moses, Y., & Ullman, S. (1997). Face recognition: The problem of compensation for changes in illumination direction. *IEEE PAMI*, *19*(4), 721-732.

Agamanolis, S., & Bove, V. M. Jr. (1997). Multi-level scripting for responsive multimedia. *IEEE Multimedia*, *4*(4), 40-50.

Basri, R., & Jacobs W. D. (2003). Lambertian reflectance and linear subspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *25*(2), 218-233.

Belhumeur, P., & Kriegman, D. (1998). What is the set of images of an object under all possible illumination conditions. *Int'l J.Computer Vision*, *28*(3), 245-260.

Black, M. J., Fleet, D. J., & Yacoob, Y. (2000). Robustly estimating changes in image appearance. *Computer Vision and Image Understanding*, *78*(1), 8-31.

Chen, S., & Lovell, B, C. (2004). Illumination and expression invariant face recognition with one sample image. *Proceedings of the International Conference on Pattern Recognition*, Cambridge, August 23-26.

Chen, S., & Lovell, B. C. (2003). Face recognition with one sample image per class. *Proceedings of ANZIIS2003*, Sydney, December 10-12, 83-88.

Cootes, T. F., Edwards G. J., & Taylor C. J. (2001). Active appearance models. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, *23*(6), 681-685.

Face Recognition Vendor Test (2006). Retrieved from http://www.frvt.org/FRVT2006/

Gao, Y., & Leung, M.K.H. (2002). Face recognition using line edge map. *IEEE PAMI*, *24*(6), 764-779.

Georghiades, A.S., Belhumeur, P.N., & Kriegman, D.J. (2001). From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *23*(6), 643-660.

Google Face Search. http://images.google.com/imghp

Hsu, W., Lee, M.L., & Zhang, J. (2002). Image mining: Trends and developments. *Journal of Intelligent Information Systems*, *19*(1), 7-23.

Liu, X. M., Chen, T., & Kumar, B.V.K.V. (2003). Face authentication for multiple subjects using eigenflow. *Pattern Recognition, Special Issue on Biometric*, *36*(2), 313-328.

Liu, C., & Wechsler, H. (1998). Evolution of optimal projection axes (OPA) for face recognition. *Third IEEE International Conference on Automatic Face and Gesture Recognition, FG'98*, Nara, Japan, April 14-16, 282-287.

Martinez, M. A. (2002). Recognizing imprecisely localized, partially occluded and expression variant faces from a single sample per class. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(6), 748-763.

Sanderson C., Shan T., & Lovell, B. C. (2007). Towards pose-invariant 2D face classification for surveillance. *The 3rd International Workshop of Analysis and Modeling of Faces and Gestures*, 276-289.

Siggelkow, S., Schael, M. & Burkhardt, H. (2001). SIMBA - Search IMages By Appearance. *Proceedings of the DAGM-Symposium on Pattern Recognition*, 9-16. Retrieved from http://simba.informatik.uni-freiburg.de/cgi-simba/SIMBA.cgi

Shan, T., Lovell, B. C., & Chen, S. (2006). Face recognition robust to head pose from one sample image. *Proceedings of 18th International Conference on Pattern Recognition*, *1*, 515-518, Hong Kong.

RIYA Visual Search. http://www.riya.com/

The Hypersoap Project. http://www.media.mit.edu/hypersoap/

Turk M. A., & Pentland, A. P. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, *3*(1), 71-86.

Yilmaz, A., & Gokmen, M. (2000). Eigenhill vs. eigenface and eigenedge. *Proceedings of International Conference Pattern Recognition*, Barcelona, Spain, 827-830.

Zhao, L., & Yang, Y. H. (1999). Theoretical analysis of illumination in PCA-based vision systems. *Pattern Recognition*, *32*, 547-564.

Zhao, W, Chellappa, R., Philips, P.J., & Rosenfeld, A. (2003). Face recognition: A literature survey. *ACM Computing Survey*, *35*, 399-458.

KEY TERMS

Biometric: A measurable, physical characteristic or personal behavioral trait used to recognize the identity, or verify the claimed identity, of an enrollee. A biometric identification system identifies a human from a measurement of a physical feature or repeatable action of the individual (for example, hand geometry, retinal scan, iris scan, fingerprint patterns, facial characteristics, DNA sequence characteristics, voice prints, and hand written signature).

Computer Vision: Using computers to analyze images and video streams and extract meaningful information from them in a similar way to the human vision system. It is related to artificial intelligence and image processing and is concerned with computer processing of images from the real world to recognize features present in the image.

Eigenfaces: Another name for face recognition via principal components analysis.

Metadata: Labeling, information describing other information.

Pattern Recognition: Pattern Recognition is the ability to take in raw data, such as images, and take action based on the category of the data.

Principal Components Analysis: Principal components analysis (PCA) is a method that can be used to simplify a dataset. It is a transform that chooses a new coordinate system for the data set, such, that that greatest variance by any projection of the data set comes to lie on the first axis (then called the first principal component), the second greatest variance on the second axis and so on. PCA can be used for reducing dimensionality. PCA is also called the Karhunen-Loève transform or the Hotelling transform.

Robust: The opposite of Brittle; this can be said of a system that has the ability to recover gracefully from the whole range of exceptional inputs and situations in a given environment. Also has the connotation of elegance in addition to careful attention to detail.

Rough Sets and Data Mining

Jerzy W. Grzymala-Busse
University of Kansas, USA

Wojciech Ziarko
University of Regina, Canada

INTRODUCTION

Discovering useful models capturing regularities of natural phenomena or complex systems until recently was almost entirely limited to finding formulae fitting empirical data. This worked relatively well in physics, theoretical mechanics, and other areas of science and engineering. However, in social sciences, market research, medicine, pharmacy, molecular biology, learning and perception, and in many other areas, the complexity of the natural processes and their common lack of analytical smoothness almost totally exclude the use of standard tools of mathematics for the purpose of data-based modeling. A fundamentally different approach is needed in those areas. The availability of fast data processors creates new possibilities in that respect. This need for alternative approaches to modeling from data was recognized some time ago by researchers working in the areas of neural nets, inductive learning, rough sets, and, more recently, data mining. The empirical models in the form of data-based structures of decision tables or rules play similar roles to formulas in classical analytical modeling. Such models can be analyzed, interpreted, and optimized using methods of rough set theory.

BACKGROUND

The theory of rough sets was originated by Pawlak (1982) as a formal mathematical theory, modeling knowledge about a universe of interest in terms of a collection of equivalence relations. Its main application areas are acquisition, analysis, and optimization of computer-processible models from data. The models can represent functional, partially functional, and probabilistic relations existing in data in the extended rough set approaches (Grzymala-Busse, 1998; Katzberg & Ziarko, 1996; Slezak & Ziarko, 2003; Ziarko, 1993). When deriving the models in the context of the rough set theory, there is no need for any additional information about data, such as, for example, probability distribution function in statistical theory, grade of membership in fuzzy set theory, and so forth (Grzymala-Busse, 1988).

The original rough set approach is concerned with investigating properties and limitations of knowledge. The main goal is forming discriminative descriptions of subsets of a universe of interest. The approach is also used to investigate and prove numerous useful algebraic and logical properties of knowledge and of approximately defined sets, called *rough sets*. The knowledge is modeled by an equivalence relation representing the ability to partition the universe into classes of indiscernible objects, referred to as *elementary sets*. The presence of the idea of approximately defined sets is a natural consequence of imperfections of existing knowledge, which may be incomplete, imprecise, or uncertain. Only an approximate description, in general, of a set (target set) can be formed. The approximate description consists of specification of lower and upper set approximations. The approximations are definable sets. The lower approximation is a union of all elementary sets contained in the target set. The upper approximation is a union of all elementary sets overlapping the target set. This ability to create approximations of non-definable, or rough, sets allows for development of approximate classification algorithms for prediction, machine learning, pattern recognition, data mining, and so forth. In these algorithms, the problem of classifying an observation into an undefinable category, which is not tractable, in the sense that the discriminating description of the category does not exist, is substituted by the problem of classifying the observation into an approximation of the category.

MAIN THRUST

The article is focused on data-mining-related extensions of the original rough set model. Based on the representative extensions, data mining techniques and applications are reviewed.

Extensions of Rough Set Theory

Developing practical applications of rough set theory revealed the limitations of this approach. For example, when dealing with market survey data, it was not possible to identify non-empty lower approximation of the target category of buyers of a product. Similarly, it often was not possible to identify non-trivial upper approximation of the target category, such as would not extend over the whole universe. These limitations follow from the fact that practical classification problems are often non-deterministic. When dealing with such problems, perfect prediction accuracy is not possible and not expected. The need to make rough set theory applicable to a more comprehensive class of practical problems inspired the development of extensions of the original approach to rough sets.

One such extension is the variable precision rough set model (VPRSM) (Ziarko, 1993). As in the original rough set theory, set approximations also are formed in VPRSM. The VPRSM criteria for forming the lower and upper approximations are relaxed, in particular by allowing a controlled degree of misclassification in the lower approximation of a target set. The resulting lower approximation represents an area of the universe where the correct classification can be made with desired probability of success, rather than deterministically. In this way, the VPRSM approach can handle a comprehensive class of problems requiring developing non-deterministic models from data. The VPRSM preserves all basic properties and algorithms of the Pawlak approach to rough sets. The algorithms are enhanced additionally with probabilistic information acquired from data (Katzberg & Ziarko, 1996; Ziarko, 1998, 2003, Ziarko & Xiao, 2004). The structures of decision tables and rules derived from data within the framework of VPRSM have probabilistic confidence factors to reflect the degree of uncertainty in classificatory decision making. The objective of such classifiers is to improve the probability of success rather than trying to guarantee 100% correct classification.

Another extension of rough set theory is implemented in the data mining system LERS (Grzymala-Busse, 1992, 1994), in which rules are equipped with three coefficients characterizing rule quality: specificity (i.e., the total number of attribute-value pairs on the left-hand side of the rule); strength (i.e., the total number of cases correctly classified by the rule during training; and the total number of training cases matching the left-hand side of the rule. For classification of unseen cases, the LERS incorporates the ideas of genetic learning, extended to use partial matching of rules and cases. The decision to which a case belongs is made on the basis of support, defined as the sum of scores of all matching rules from the class, where a score of the rule is the product of the first two coefficients associated with the rule. As indicated by experiments, partial matching is a valuable mechanism when complete matching fails (Grzymala-Busse, 1994). In the LERS classification system, the user may use 16 strategies for classification. In some of these strategies, the final decision is based on probabilities acquired from raw data (Grzymala-Busse & Zou, 1998).

Other extensions of rough set theory include generalizations of the basic concept of rough set theory—the indiscernibility relation. A survey of such methods was presented in Yao (2003).

From Data to Rough Decision Tables

When deriving models from data within the rough set framework, one of the primary constructs is a decision table derived from data referred to as *rough decision table* (Pawlak, 1991; Ziarko, 1999, 2002a). The rough decision table represents knowledge about the universe of interest and the relation between the knowledge and the target set or sets. The idea of the rough decision table was formulated in both the original framework of rough sets and in the extended VPRSM. In the latter case, the table is called *probabilistic decision table* (Ziarko, 2002a). In the table, some columns correspond to descriptive attributes used to classify objects of the domain of interest, while other columns represent target sets or rough approximations of the sets. The rows of the table represent the classes of the classification of the domain in terms of the descriptive attributes. If the decision table contains representatives of all or almost all classes of the domain, and if the relation with the prediction targets is completely or almost completely

specified, then the table can be treated as a model of the domain. Such a model represents descriptions of all or almost all objects of the domain and their relationship to the prediction target. The specification of the relationship may include empirical assessments of conditional probabilities, if the VPRSM approach is used in model derivation. If the model is complete enough, and if the data-based estimates of probabilities are relatively close to real values, then the decision table can be used as a basis of a classifier system. To ensure relative completeness and generality of the decision table model, the values of the attributes used to construct the classification of the domain need to be sufficiently general. For example, in many practical problems, rather than using precise numeric measurements, value ranges often are used after preliminary discretization of original precise values. This conversion of original data values into secondary, less precise representation is one of the major pre-processing steps in rough set-based methodology. The acquired decision table can be further analyzed and optimized using classical algorithms for interattribute dependency computation and minimal nonredundant subset of attributes (attribute reduct) identification (Pawlak, 1991; Ziarko 2002b).

From Data to Rule Sets

A number of systems for machine learning and data mining have been developed in the course of research on theory and applications of rough sets (Grzymala-Busse, 1992; Ohrn & Komorowski, 1997; Ziarko, 1998b; Ziarko et al., 1993). The representative example of such developments is the data mining system LERS, whose first version was developed at the University of Kansas in 1988. The current version of LERS is essentially a family of data mining systems. The main objective of LERS is computation of decision rules from data. Computed rule sets may be used for classification of new cases or for interpretation of knowledge. The LERS system may compute rules from imperfect data (Grzymala-Busse, 1992) (e.g., data with missing attribute values or inconsistent cases). LERS is also equipped with a set of discretization schemas to deal with numerical attributes. In addition, a variety of LERS methods may help to handle missing attribute values. LERS accepts inconsistent input data (i.e., characterized by the same values of all attributes, but belonging to two different target sets). For inconsistent data, LERS computes lower and upper approximations of all sets involved. The system is also assisted with tools for rule validation, such as leaving-one-out, 10-fold cross validation, and holdout.

LERS has proven its applicability having been used for two years by NASA Johnson Space Center (Automation and Robotics Division) as a tool to develop expert systems of the type most likely to be used in medical decision making on the board of the International Space Station. LERS also was used to enhance facility compliance under Sections 311, 312, and 313 of Title III of the Emergency Planning and Community Right to Know (Grzymala-Busse, 1993). The LERS system was used in other areas, as well (e.g., in the medical field to compare the effects of warming devices for postoperative patients, to assess preterm birth) (Woolery & Grzymala-Busse, 1994) and for diagnosis of melanoma (Grzymala-Busse et al., 2001).

FUTURE TRENDS

The literature related to the subject of rough sets exceeds well over 1,000 publications. By necessity, in what follows, we cite only some representative examples of the research works on the subject. A comprehensive up-to-date collection of references can be found online at http://rsds.wsiz.rzeszow.pl (Suraj, 2004).

Following Pawlak's original publication (Pawlak, 1991), the mathematical fundamentals of the original rough set model were published in Polkowski (2002). There exits an extensive body of literature on rough set theory applications to knowledge discovery and data mining. In particular, a comprehensive review is available in Polkowski and Skowron (1998). The basic algorithms for data mining applications using the original rough set theory were summarized in Ziarko (2002b). Since the introduction of the original RST, several extensions of the original model were proposed (Greco et al., 2000; Slezak & Ziarko, 2003; Yao & Wong 1992; Ziarko, 1993). In particular, VPRSM was published for the first time in Ziarko (1993) and was further investigated in Kryszkiewicz (1994), Beynon (2000), Slezak and Ziarko (2003), and others, and served as a basis of a novel approach to inductive logic programming (Mahesvari et al., 2001). The probabilistic decision tables were introduced in Ziarko (1998b). The LERS system was first described in Grzymala-Busse (1992). Its most important algorithm, LEM2, was presented in Chan and Grzymala-Busse (1994). Some

applications of LERS were published in Freeman, et al. (2001), Gunn and Grzymala-Busse (1994), Grzymala-Busse et al. (2001), Grzymala-Busse and Gunn (1995), Grzymala-Busse and Woolery (1994), Loupe, et al. (2001), Moradi, et al. (1995), and Woolery, et al. (1991).

It appears that utilizing extensions of the original rough set theory is the main trend in data mining applications of this approach. In particular, a number of sources reported experiments using rough set theory for medical diagnosis, control, and pattern recognition, including speech recognition, handwriting recognition, and music fragment classification (Brindle & Ziarko, 1999; Kostek, 1998; Mrozek, 1986; Peters et al., 1999; Plonka & Mrozek, 1995; Shang & Ziarko, 2003). These technologies are far from maturity, which indicates that the trend toward developing applications based on extensions of rough set theory will continue.

CONCLUSION

Data mining and machine learning applications based on the original approach to rough set theory and, more recently, on extensions and generalizations of rough set theory, have been attempted for about 20 years now. Due to space limits, this article mentions only example experimental and real-life application projects. The projects confirm the viability of rough set theory as a fundamental framework for data mining, machine learning, pattern recognition, and related application areas, and provide inspiring feedback toward continuing growth of the rough set approach to better suit the needs of real-life application problems.

REFERENCES

Beynon, M. (2000). An investigation of beta-reduct selection within variable precision rough sets model. *Proceedings of the 2nd International Conference on Rough Sets and Current Trends in Computing*, Banff, Canada.

Brindle, D., & Ziarko, W. (1999). Experiments with rough set approach to speech recognition. *Proceedings of the International Conference on Methodologies for Intelligent Systems*, Warsaw, Poland.

Chan, C.C., & Grzymala-Busse, J.W. (1994). On the two local inductive algorithms: PRISM and LEM2. *Foundations of Computing and Decision Sciences, 19*, 185-203.

Freeman, R.L., Grzymala-Busse, J.W., Laura, A., Riffel, L.A., & Schroeder, S.R. (2001). Analysis of self-injurious behavior by the LERS data mining system. *Proceedings of the Japanese Society for AI, International Workshop on Rough Set Theory and Granular Computing, RSTGC-2001*, Shimane, Japan.

Greco, S., Matarazzo, B., Slowinski, R., & Stefanowski, J. (2000). Variable consistency model of dominance-based rough sets approach. *Proceedings of the 2nd International Conference on Rough Sets*, Banff, Canada.

Grzymala-Busse, J.P., Grzymala-Busse, J.W., & Hippe, Z.S. (2001). Melanoma prediction using data mining system LERS. *Proceedings of the 25th Anniversary Annual International Computer Software and Applications Conference COMPSAC 2001*, Chicago, Illinois.

Grzymala-Busse, J.W. (1992). LERS—A system for learning from examples based on rough sets. In R. Slowinski (Ed.), *Intelligent decision support: Handbook of applications and advances of the rough sets theory.* Kluwer.

Grzymala-Busse, J.W. (1993). ESEP: An expert system for environmental protection. *Proceedings of the RSKD–93, International Workshop on Rough Sets and Knowledge Discovery*, Banff, Canada.

Grzymala-Busse, J.W. (1994). Managing uncertainty in machine learning from examples. *Proceedings of the Third Intelligent Information Systems Workshop*, Wigry, Poland.

Grzymala-Busse, J.W., & Werbrouck, P. (1998). On the best search method in the LEM1 and LEM2 algorithms. In E. Orlowska (Ed.), *Incomplete information: Rough set analysis.* Physica-Verlag.

Grzymala-Busse, J.W., & Zou, X. (1998). Classification strategies using certain and possible rules. *Proceedings of the First International Conference on Rough Sets and Current Trends in Computing*, Warsaw, Poland.

Gunn, J.D., & Grzymala-Busse, J.W. (1994). Global temperature stability by rule induction: An interdisciplinary bridge. *Human Ecology, 22*, 59-81.

Katzberg, J., & Ziarko, W. (1996). Variable precision extension of rough sets. *Fundamenta Informaticae, Special Issue on Rough Sets, 27,* 155-168.

Kostek, B. (1998). Computer-based recognition of musical phrases using the rough set approach. *Journal of Information Sciences, 104*, 15-30.

Kryszkiewicz, M. (1994). *Knowledge reduction algorithms in information systems* [doctoral thesis]. Warsaw, Poland: Warsaw University of Technology.

Loupe, P.S., Freeman, R.L., Grzymala-Busse, J.W., & Schroeder, S.R. (2001). Using rule induction for prediction of self-injuring behavior in animal models of development disabilities. *Proceedings of the 14th IEEE Symposium on Computer-Based Medical Systems*, Bethesda, Maryland.

Maheswari, U., Siromoney, A., Mehata, K., & Inoue, K. (2001). The variable precision rough set inductive logic programming model and strings. *Computational Intelligence, 17*, 460-471.

Moradi, H. et al. (1995). Entropy of English text: Experiments with humans and a machine learning system based on rough sets. *Proceedings of the 2nd Annual Joint Conference on Information Sciences*, Wrightsville Beach, North Carolina.

Mrozek, A. (1986). Use of rough sets and decision tables for implementing rule-based control of industrial processes. *Bulletin of the Polish Academy of Sciences, 34*, 332-356.

Ohrn, A., & Komorowski, J. (1997). *ROSETTA*: A rough set toolkit for analysis of data. *Proceedings of the Third International Joint Conference on Information Sciences, Fifth International Workshop on Rough Sets and Soft Computing*, Durham, North Carolina.

Pawlak, Z. (1982). Rough sets. *International Journal of Computer and Information Sciences, 11*, 341-356.

Pawlak, Z. (1984). *International Journal Man-Machine Studies, 20*, 469.

Pawlak, Z. (1991). *Rough sets: Theoretical aspects of reasoning about data*. Kluwer.

Pawlak, Z., Grzymala-Busse, J.W., Slowinski, R., & Ziarko, W. (1995). Rough sets. *Communications of the ACM, 38*, 89-95.

Peters, J., Skowron, A., & Suraj, Z. (1999). An application of rough set methods in control design. *Proceedings of the Workshop on Concurrency*, Warsaw, Poland.

Plonka, L., & Mrozek, A. (1995). Rule-based stabilization of the inverted pendulum. *Computational Intelligence, 11*, 348-356.

Polkowski, L. (2002). *Rough sets: Mathematical foundations*. Springer Verlag.

Polkowski, L., & Skowron, A. (Eds.). (1998). *Rough sets in knowledge discovery, 2, applications, case studies and software systems*. Heidelberg: Physica Verlag.

Shang, F., & Ziarko, W. (2003). Acquisition of control algorithms. *Proceedings of the International Conference on New Trends in Intelligent Information Processing and Web Mining*, Zakopane, Poland.

Slezak, D., & Ziarko, W. (2003). Variable precision Bayesian rough set model. *Proceedings of the 9th International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing*, Chongqing, China.

Slowinski, R. (Ed.). (1992). *Decision support by experience: Rough sets approach*. Kluver.

Suraj, Z., & Grochowalski, P. (2004). The rough set data base system: An overview. *Proceedings of the International Conference on Rough Sets and Current Trends in Computing*, Uppsala, Sweden.

Tsumoto, S. (2003). Extracting structure of medical diagnosis: Rough set approach. *Proceedings of the 9th International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing*, Chongqing, China.

Woolery, L., Grzymala-Busse, J., Summers, S., & Budihardjo, A. (1991). The use of machine learning program LERS_LB 2.5 in knowledge acquisition for expert system development in nursing. *Computers in Nursing, 9*, 227-234.

Yao, Y.Y. (2003). On generalizing rough set theory. *Proceedings of the 9th International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing*, Chongqing, China.

Yao, Y.Y., & Wong, S.K.M. (1992). A decision theoretic framework for approximating concepts. *Proceedings of the International Journal of Man-Machine Studies*.

Ziarko, W. (1993). Variable precision rough sets model. *Journal of Computer and Systems Sciences, 46,* 39-59.

Ziarko, W. (1998a). Approximation region-based decision tables. *Proceedings of the International Conference on Rough Sets and Current Trends in Computing,* Warsaw, Poland.

Ziarko, W. (1998b). KDD-R: Rough sets-based data mining system. In L. Polkowski, & A. Skowron (Eds.), *Rough sets in knowledge discovery, Part II* (pp. 598-601). Springer Verlag.

Ziarko, W. (2002a). Acquisition of hierarchy-structured probabilistic decision tables and rules from data. *Proceedings of the IEEE International Conference on Fuzzy Systems,* Honolulu, Hawaii.

Ziarko, W. (2002b). Rough set approaches for discovery of rules and attribute dependencies. In W. Kloesgen, & J. Zytkow (Eds.), *Handbook of data mining and knowledge discovery* (pp. 328-339). Oxford University Press.

Ziarko, W., Golan, R., & Edwards, D. (1993). An application of datalogic/R knowledge discovery tool to identify strong predictive rules in stock market data. *Proceedings of the AAAI Workshop on Knowledge Discovery in Databases,* Washington, D.C.

Ziarko, W., & Xiao, X. (2004). Computing minimal probabilistic rules from probabilistic decision tables: Decision matrix approach. *Proceedings of the Atlantic Web Intelligence Conference,* Cancun, Mexico.

KEY TERMS

Decision Rule: Specification of the relationship between collection of observations (conditions) and an outcome (a decision).

Definable Set: A set that has a description precisely discriminating elements of the set from among all elements of the universe of interest.

LERS: A comprehensive system for data mining based on rough sets.

Lower Approximation of a Rough Set: Maximum definable set contained in the rough set.

Rough Decision Table: Collection of disjoint decision rules of identical format.

Rough Set: An undefinable set.

Upper Approximation of a Rough Set: Minimum definable set containing the rough set.

Variable Precision Rough Set Model: An approach to forming lower and upper approximations of a rough set via generalized parametric definitions.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 973-977, copyright 2005 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

Sampling Methods in Approximate Query Answering Systems

Gautam Das
The University of Texas at Arlington, USA

INTRODUCTION

In recent years, advances in data collection and management technologies have led to a proliferation of very large databases. These large data repositories typically are created in the hope that, through analysis such as data mining and decision support, they will yield new insights into the data and the real-world processes that created them. In practice, however, while the collection and storage of massive datasets has become relatively straightforward, effective data analysis has proven more difficult to achieve. One reason that data analysis successes have proven elusive is that most analysis queries, by their nature, require aggregation or summarization of large portions of the data being analyzed. For multi-gigabyte data repositories, this means that processing even a single analysis query involves accessing enormous amounts of data, leading to prohibitively expensive running times. This severely limits the feasibility of many types of analysis applications, especially those that depend on timeliness or interactivity.

While keeping query response times short is very important in many data mining and decision support applications, exactness in query results is frequently less important. In many cases, ballpark estimates are adequate to provide the desired insights about the data, at least in preliminary phases of analysis. For example, knowing the marginal data distributions for each attribute up to 10% error often will be enough to identify top-selling products in a sales database or to determine the best attribute to use at the root of a decision tree.

For example, consider the following SQL query:

```
SELECT State, COUNT(*) as ItemCount
FROM SalesData
WHERE ProductName= 'LawnMower'
GROUP BY State
ORDER BY ItemCount DESC
```

This query seeks to compute the total number of a particular item sold in a sales database, grouped by state. Instead of a time-consuming process that produces completely accurate answers, in some circumstances, it may be suitable to produce ballpark estimates (e.g., counts to the nearest thousands).

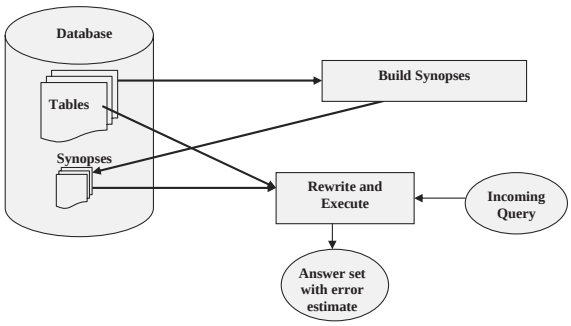
The acceptability of inexact query answers, coupled with the necessity for fast query response times, has led researchers to investigate approximate query answering (AQA) techniques that sacrifice accuracy to improve running time, typically through some sort of lossy data compression. The general rubric in which most approximate query processing systems operate is as follows: first, during the preprocessing phase, some auxiliary data structures, or data synopses, are built over the database; then, during the runtime phase, queries are issued to the system and approximate query answers quickly are returned, using the data synopses built during the preprocessing phase. The quality of an approximate query processing system often is determined by how accurately the synopsis represents the original data distribution, how practical it is to modify existing database systems to incorporate approximate query answering, and whether error estimates can be returned in addition to ballpark estimates.

BACKGROUND

Figure 1 describes a general architecture for most AQA systems. There are two components in the architecture: (1) a component for building the synopses from database relations, and (2) a component that rewrites an incoming query in order to use the synopses to answer the query approximately and report the answer with an estimate of the error in the answer.

The different approximate query answering systems that have been proposed differ in various ways: in the types of synopses proposed; whether the synopses building component is executed in a preprocessing phase or whether it executes at runtime; the ability of

Figure 1. Architecture for approximate query answering



the AQA system also to provide error guarantees in addition to the approximate answers; and, finally (from a practical point of view and perhaps the most important), the amount of changes necessary to query processing engines of commercial database management systems to incorporate approximate query answering.

The types of synopses developed for AQA systems can be divided into two broad groups: sampling-based approaches and non-sampling-based approaches. In sampling-based approaches, a small random sample of the rows of the original database table is prepared, and queries are directed against this small sample table. The non-sampling-based approaches encompass a wide variety of techniques (e.g., sophisticated data structures such as wavelets [Chakrabarti et al., 2001; Matias, Vitter & Wang, 1998] and histograms [Ioannidis & Poosala, 1999]) have been proposed as useful tools for AQA.

Work in non-sampling-based AQA techniques is of great theoretical interest, but its practical impact often is limited by the extensive modifications to query processors and query optimizers that often are needed to make use of these technologies. On the other hand, sampling-based systems have the advantage that they can be implemented as a thin layer of middleware that rewrites queries to run against sample tables stored as ordinary relations in a standard, off-the-shelf database server.

Partly for these reasons, sampling-based systems have in recent years been the most heavily studied type of AQA system. In the rest of this article, our focus is on presenting an overview of the latest developments in sampling-based AQA techniques.

MAIN THRUST

In the following section, we summarize the various sampling-based AQA technologies that have been proposed in recent years by the research community. The focus of this article is on approximately answering standard SQL queries on relational databases; other exciting work done on approximate query processing in other scenarios, such as streaming and time series data, is beyond the scope of this article.

We assume a standard data warehouse schema, consisting of a few fact tables containing the measure columns, connected to several dimension tables via foreign key relationships. Furthermore, we assume that our queries are aggregation queries with SUM, COUNT, and GROUP BY operators, either over a single fact table or over a fact table joined to several dimension tables.

Uniform Random Sampling

The essential idea is that a small precomputed uniform random sample of rows S of the database R often represents the entire database well. For a fast approximate answer at runtime, one simply has to execute the query on S and scale the result. Thus, if S is a 1% sample of the database, the scaling factor is 100. The main advantages of uniform random sampling are simplicity and efficiency of preprocessing. However, there are several critical disadvantages that have not allowed this approach to be considered seriously for AQA systems.

One disadvantage is the well-known statistical problem of large data variance. For example, suppose we wish to estimate the average salaries of a particular corporation. Uniform random sampling does badly if the salary distribution is highly skewed.

The other disadvantage is specific to database systems, and is the low selectivity problem. For example, suppose a query wishes to find the average salary of a small department of a large corporation. If we only had a uniform random sample of the entire database, then it is quite likely that this small department may not be adequately represented, leading to large errors in the estimated average salary.

To mitigate these problems, much research has been attempted using so-called biased sampling techniques, where a non-uniform random sample is precomputed, such that parts of the database deemed more important

than the rest are better represented in the sample. We discuss such techniques later in the article.

Online Aggregation

Hellerstein, Haas, and Wang (1997) describe techniques for online aggregation in which approximate answers for queries are produced during early stages of query processing and gradually refined until all the data have been processed. This framework is extended in Raman and Hellerstein (2002) to have the query processor give precedence to tuples that contribute to higher-priority parts of the query result, where priority is defined using a user-specified function. The online aggregation approach has some compelling advantages (e.g., it does not require preprocessing, and it allows progressive refinement of approximate answers at runtime until the user is satisfied or the exact answer is supplied, and it can provide confidence intervals that indicate the uncertainty present in the answer).

However, there are two important systems considerations that represent practical obstacles to the integration of online aggregation into conventional database systems. First, stored relations are frequently clustered by some attribute, so accessing tuples in a random order, as required for online aggregation, requires (slow) random disk accesses. Second, online aggregation necessitates significant changes to the query processor of the database system. This is impractical, as it is desirable for an AQA system to leverage today's commercial query processing systems with minimal changes to the greatest degree possible.

Next, we consider several biased-sampling AQA methods that are based on precomputing the samples. Toward the end, we also discuss a method that attempts to strike a balance between online and precomputed sampling.

Icicles

Recognizing the low selectivity problem, designing a biased sample that is based on known workload information was attempted in Ganti, Lee, and Ramakrishnan (2000). In this paper, the assumption was that a workload of queries (i.e., a log of all recent queries executing against the database) is a good predictor of the queries that are yet to execute on the database in the future. Thus, for example, if a query requests for the average salary of a small department in a large corporation, it is assumed that such (or similar) queries will repeat in the future. A heuristic precomputation procedure called *Icicles* was developed, in which tuples that have been accessed by many queries in the workload were assigned greater probabilities of being selected into the sample.

While this was an interesting idea based on biased sampling that leverages workload information, a disadvantage was that it focuses only on the low selectivity problem, and, furthermore, the suggested solution is rather heuristical.

Outlier Indexing

The first paper that attempted to address the problem of large data variance was by Chaudhuri, Das, Datar, Motwani, and Narasayya (2001). It proposes a technique called *Outlier Indexing* for improving sampling-based approximations for aggregate queries, when the attribute being aggregated has a skewed distribution.

The basic idea is that outliers of the data (i.e., the records that contribute to high variance in the aggregate column) are collected into a separate index, while the remaining data is sampled using a biased sampling technique. Queries are answered by running them against both the outlier index as well as the biased sample, and an estimated answer is composed out of both results. A disadvantage of this approach was that the primary emphasis was on the data variance problem, and while the authors did propose a hybrid solution for both the data variance as well as the low selectivity problem, the proposed solution was heuristical and, therefore, suboptimal.

Congressional Sampling

The AQUA project at Bell Labs (Acharya, Gibbons & Poosala, 1999) developed a sampling-based system for approximate query answering. Techniques used in AQUA included congressional sampling (Acharya, Gibbons & Poosala, 2000), which is targeted toward answering a class of common and useful analysis queries (group by queries with aggregation). Their approach stratifies the database by considering the set of queries involving all possible combinations of grouping columns and produces a weighted sample that balances the approximation errors of these queries. However, their approach is still ad hoc in the sense that even though they try to reduce the error, their scheme

does not minimize the error for any of the well-known error metrics.

Join Synopses

The AQUA project at Bell Labs also developed the join synopses technique (Acharya et al., 1999), which allows approximate answers to be provided for certain types of join queries; in particular, foreign-key joins. The technique involved precomputing the join of samples of fact tables with dimension tables, so that at runtime, queries only need to be executed against single (widened) sample tables. This is an alternate to the approach of only precomputing samples of fact tables and having to join these sample tables with dimension tables at runtime.

We mention that the problem of sampling over joins that are not foreign-key joins is a difficult problem and, under certain conditions, is essentially not possible (Chaudhuri, Motwani & Narasayya, 1999). Thus, approximate query answering does not extend to queries that involve non-foreign key joins.

Stratified Sampling (STRAT)

The paper by Chaudhuri, Das, and Narasayya (2001) sought to overcome many of the limitations of the previous works on precomputed sampling for approximate query answering and proposed a method called STRAT for approximate query answering.

Unlike most previous sampling-based studies that used ad-hoc randomization methods, the authors here formulated the problem of precomputing a sample as an optimization problem, whose goal is to minimize the error for the given workload. They also introduced a generalized model of the workload (lifted workload) that makes it possible to tune the selection of the sample, so that approximate query processing using the sample is effective, not only for workloads that are exactly identical to the given workload, but also for workloads that are similar to the given workload (i.e., queries that select regions of the data that overlap significantly with the data accessed by the queries in the given workload)—a more realistic scenario. The degree of similarity can be specified as part of the user/database administrator preference. They formulate selection of the sample for such a lifted workload as a stratified sampling task with the goal to minimize error in estimation of aggregates. The benefits of this systematic approach are demonstrated by theoretical results (where it is shown to subsume much of the previous work on precomputed sampling methods for AQA) and experimental results on synthetic data as well as real-enterprise data warehouses.

Dynamic Sample Selection

A sampling technique that attempts to strike a middle ground between precomputed and online sampling is dynamic sample selection (Babcock, Chaudhuri & Das, 2003).

The requirement for fast answers during the runtime phase means that scanning a large amount of data to answer a query is not possible, or else the running time would be unacceptably large. Thus, most sampling-based approximate query answering schemes have restricted themselves to building only a small sample of the data. However, because relatively large running times and space usage during the preprocessing phase are generally acceptable, as long as the time and space consumed are not exorbitant, nothing prevents us from scanning or storing significantly larger amounts of data during preprocessing than we are able to access at runtime. Of course, because we only are able to access a small amount of stored data at runtime, there is no gain to be had from building large auxiliary data structures, unless they are accompanied by some indexing technique that allows us to decide, for a given query, which (small) portion of the data structures should be accessed to produce the most accurate approximate query answer.

In Babcock, Chaudhuri, and Das (2003), the authors describe a general system architecture for approximate query processing that is based on the dynamic sample selection technique. The basic idea is to construct during the preprocessing phase a random sample containing a large number of differently biased subsamples, and then, for each query that arrives during the runtime phase, to select an appropriate small subset from the sample that can be used to give a highly accurate approximate answer to the query. The philosophy behind dynamic sample selection is to accept greater disk usage for summary structures than other sampling-based AQA methods in order to increase accuracy in query responses while holding query response time constant (or, alternatively, to reduce query response time while

holding accuracy constant). The belief is that for many AQA applications, response time and accuracy are more important considerations than disk usage.

FUTURE TRENDS

In one sense, AQA systems are not new. These methods have been used internally for a long time by query optimizers of database systems for selectivity estimation. However, approximate query answering has not been externalized yet to the end user by major vendors, though sampling operators are appearing in commercial database management systems. Research prototypes exist in the industry (e.g., AQP from Microsoft Research and the AQUA system from Bell Labs).

From a research potential viewpoint, approximate query answering promises to be a very fertile area with several deep and unresolved problems. Currently, there is a big gap between the development of algorithms and their adaptability in real systems. This gap needs to be addressed before AQA techniques can be embraced by the industry. Second, the research has to broaden beyond the narrow confines of aggregation queries over single table databases or multi-tables involving only foreign-key joins. It is important to investigate how to return approximations to set-valued results, AQA over multi-table databases with more general types of SQL queries, AQA over data streams, and investigations into the practicality of other non-sampling-based approaches to approximate query answering. As data repositories get larger and larger, effective data analysis will prove increasingly more difficult to accomplish.

CONCLUSION

In this article, we discussed the problem of approximate query answering in database systems, especially in decision support applications. We described various approaches taken to design approximate query answering systems, especially focusing on sampling-based approaches. We believe that approximate query answering is an extremely important problem for the future, and much work needs to be done before practical systems can be built that leverage the substantial theoretical developments already accomplished in the field.

REFERENCES

Acharya, S. et al. (1999). Join synopses for approximate query answering. *Proceedings of the Special Interest Group on Management of Data*.

Acharya, S., Gibbons, P.B., & Poosala, V. (1999). Aqua: A fast decision support system using approximate query answers. *Proceedings of the International Conference on Very Large Databases*.

Acharya, S., Gibbons, P.B., & Poosala, V. (2000). Congressional samples for approximate answering of group-by queries. *Proceedings of the Special Interest Group on Management of Data*.

Babcock, B., Chaudhuri, S., & Das, G. (2003). Dynamic sample selection for approximate query processing. *Proceedings of the Special Interest Group on Management of Data*.

Chakrabarti, K., Garofalakis, M.N., Rastogi, R., & Shim, K. (2001). Approximate query processing using wavelets. *Proceedings of the International Conference on Very Large Databases*.

Chaudhuri,S., Das, G., Datar, M., Motwani, R., & Narasayya, V. (2001). Overcoming limitations of sampling for aggregation queries. *Proceedings of the International Conference on Data Engineering*.

Chaudhuri, S., Das, G., & Narasayya, V. (2001). A robust, optimization-based approach for approximate answering of aggregate queries. *Proceedings of the Special Interest Group on Management of Data*.

Chaudhuri, S., Motwani, R., & Narasayya, V. (1999). On random sampling over joins. *Proceedings of the Special Interest Group on Management of Data*.

Ganti, V., Lee, M., & Ramakrishnan, R. (2000). ICICLES: Self-tuning samples for approximate query answering. *Proceedings of the International Conference on Very Large Databases*.

Hellerstein, J.M., Haas, P.J., & Wang, H. (1997). Online aggregation. *Proceedings of the Special Interest Group on Management of Data*.

Ioannidis, Y.E., & Poosala, V. (1999). Histogram-based approximation of set-valued query-answers. *Proceedings of the International Conference on Very Large Databases*.

Matias, Y., Vitter, J.S., & Wang, M. (1998). Wavelet-based histograms for selectivity estimation. *Proceedings of the Special Interest Group on Management of Data*.

Raman, V., & Hellerstein, J.M. (2002). Partial results for online query processing. *Proceedings of the Special Interest Group on Management of Data*.

KEY TERMS

Aggregation Queries: Common queries executed by decision support systems that aggregate and group large amounts of data, where aggregation operators are typically SUM, COUNT, AVG, and so forth.

Biased Sampling: A random sample of k tuples of a database, where the probability of a tuple belonging to the sample varies across tuples.

Decision Support Systems: Typically, business applications that analyze large amounts of data in warehouses, often for the purpose of strategic decision making.

Histograms: Typically used for representing one-dimensional data, though multi-dimensional histograms are being researched in the database field. A histogram is a division of the domain of a one-dimensional ordered attribute into buckets, where each bucket is represented by a contiguous interval along the domain, along with the count of the number of tuples contained within this interval and other statistics.

Standard Error: The standard deviation of the sampling distribution of a statistic. In the case of approximate query answering, it measures the expected value of the error in the approximation of aggregation queries.

Stratified Sampling: A specific procedure for biased sampling, where the database is partitioned into different strata, and each stratum is uniformly sampled at different sampling rates. Tuples that are more important for aggregation purposes, such as outliers, are put into strata that are then sampled at a higher rate.

Uniform Sampling: A random sample of k tuples of a database, where each subset of k tuples is equally likely to be the sample.

Workload: The log of all queries that execute on a database system. Workloads often are used by database administrators as well as by automated systems (such as AQA systems) to tune various parameters of database systems for optimal performance, such as indexes and physical design, and, in the case of AQA, the set of sample tables.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 990-994, copyright 2005 by IGI Publishing, formerly known as Idea Group Publishing (an imprint of IGI Global).

Scalable Non-Parametric Methods for Large Data Sets

V. Suresh Babu
Indian Institute of Technology-Guwahati, India

P. Viswanath
Indian Institute of Technology-Guwahati, India

M. Narasimha Murty
Indian Institute of Science, India

INTRODUCTION

Non-parametric methods like the nearest neighbor classifier (NNC) and the Parzen-Window based density estimation (Duda, Hart & Stork, 2000) are more general than parametric methods because they do not make any assumptions regarding the probability distribution form. Further, they show good performance in practice with large data sets. These methods, either explicitly or implicitly estimates the probability density at a given point in a feature space by counting the number of points that fall in a small region around the given point. Popular classifiers which use this approach are the NNC and its variants like the k-nearest neighbor classifier (k-NNC) (Duda, Hart & Stock, 2000). Whereas the DBSCAN is a popular density based clustering method (Han & Kamber, 2001) which uses this approach. These methods show good performance, especially with larger data sets. Asymptotic error rate of NNC is less than twice the Bayes error (Cover & Hart, 1967) and DBSCAN can find arbitrary shaped clusters along with noisy outlier detection (Ester, Kriegel & Xu, 1996).

The most prominent difficulty in applying the non-parametric methods for large data sets is its computational burden. The space and classification time complexities of NNC and k-NNC are $O(n)$ where n is the training set size. The time complexity of DBSCAN is $O(n^2)$. So, these methods are not scalable for large data sets. Some of the remedies to reduce this burden are as follows. (1) Reduce the training set size by some editing techniques in order to eliminate some of the training patterns which are redundant in some sense (Dasarathy, 1991)**.** For example, the condensed NNC (Hart, 1968) is of this type. (2) Use only a few selected prototypes from the data set. For example, *Leaders-subleaders* method and *l*-DBSCAN method are of this type (Vijaya, Murthy & Subramanian, 2004 and Viswanath & Rajwala, 2006). These two remedies can reduce the computational burden, but this can also result in a poor performance of the method. Using enriched prototypes can improve the performance as done in (Asharaf & Murthy, 2003) where the prototypes are derived using adaptive rough fuzzy set theory and as in (Suresh Babu & Viswanath, 2007) where the prototypes are used along with their relative weights.

Using a few selected prototypes can reduce the computational burden. Prototypes can be derived by employing a clustering method like the leaders method (Spath, 1980)**,** the *k*-means method (Jain, Dubes, & Chen, 1987), *etc.*, which can find a partition of the data set where each block (cluster) of the partition is represented by a prototype called leader, centroid, *etc*. But these prototypes can not be used to estimate the probability density, since the density information present in the data set is lost while deriving the prototypes. The chapter proposes to use a modified leader clustering method called the *counted-leader* method which along with deriving the leaders preserves the crucial density information in the form of a *count* which can be used in estimating the densities. The chapter presents a fast and efficient nearest prototype based classifier called the *counted k-nearest leader classifier (ck-NLC)* which is on-par with the conventional k-NNC, but is considerably faster than the k-NNC. The chapter also presents a density based clustering method called *l*-DBSCAN which is shown to be a faster and scalable version of DBSCAN (Viswanath & Rajwala, 2006). Formally, under some assumptions, it is shown that the number of leaders is upper-bounded by a constant which is independent of the data set size and the distribution from which the data set is drawn.

BACKGROUND

Supervised learning and *unsupervised learning* are two main paradigms of learning. Supervised learning refers to learning with a teacher, typically in situations where one has a set of training patterns whose class labels are known. The objective is to assign a label to the given test pattern. This is called pattern classification. On the other hand, unsupervised learning or learning without a teacher refers to situations where training patterns are not labeled and the typical objective is to find the natural grouping (or categories) among the given patterns. This is called pattern clustering (Jain, Murty & Flynn, 1999). Among various classification and clustering methods non-parametric methods are those which either explicitly or implicitly estimates the arbitrary density function from the data sets based on which classification or clustering tasks can be performed. Prominent non-parametric classifiers are NNC and k-NNC. Whereas DBSCAN is a popular non-parametric density based clustering method.

NNC works as follows. Let $\{(X^1, y^1),\ldots,(X^n, y^n)\}$ be the training set where y^i is the class label for the pattern X^i, for $1 \leq i \leq n$. For a given test pattern T, Let X^l be the nearest neighbor in the training set based on the given distance measure, then NNC assigns the class label of X^l *(i.e.,*y^l*)* to T. An extension of the above method is to find k nearest neighbors of the test pattern and assigning the class label to the test pattern based on a majority vote among the k neighbors. Assuming that k is a small constant when compared with n, the time required to classify a pattern, either by NNC or by k-NNC, is $O(n)$.

Density based clustering methods like DBSCAN groups the data points which are dense and connected into a single cluster. Density at a point is found non-parametrically. It is assumed that probability density over a small region is uniformly distributed and the density is given by m/nV, where m is the number of points out of n input data points that are falling in a small region around the point and V is the volume of the region. The region is assumed to be a hyper sphere of radius ε and hence threshold density can be specified by a parameter *MinPts*, the minimum number of points required to be present in the region to make it dense. Given an input dataset D, and the parameters ε and *MinPts*, DBSCAN finds a dense point in D and expands it by merging neighboring dense points. Patterns in the data set which do not belong to any of the clusters are called noisy patterns. A non dense point can be a part of a cluster if it is at distance less than or equal to ε from a dense pattern, otherwise it is a noisy outlier (Viswanath & Rajwala, 2006). The time complexity of DBSCAN is $O(n^2)$.

Many non-parametric methods suffer from the huge computational requirements and are not scalable to work with large data sets like those in data mining applications.

MAIN FOCUS

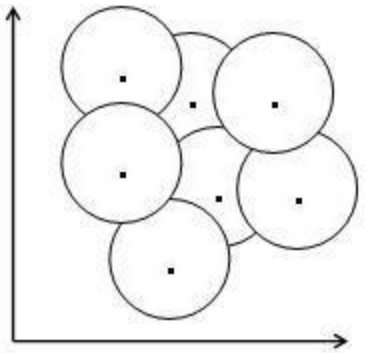
Using only a few selected prototypes from the data set can reduce the computational burden of the non-parametric methods. The prototypes need to be rich enough to compute the probability density at an arbitrary point in the feature space by using them. First, *counted-leader* method is described followed by *counted k-nearest leader classifier* and a hybrid density based clustering method which uses the counted prototypes in place of the large training data set. Finally, some experimental results are given in support of the methods in this chapter.

Counted-Leader Method

Counted-leader method which is a modified leader clustering method scans the database only once and is an incremental method. It has running time that is linear in the size of the input data set. More precisely, it can find a partition of the data set in $O(n)$ time where n is the data set size. It derives prototypes and also a count for each prototype which indicates the prototypes relative importance. The count of a prototype represents the number of patterns falling under that prototype. Conventional leader method (Spath, 1980) derives the prototypes called the leaders set. These leaders represent the semi-spherical clusters as shown in the Figure 1, and are more or less uniformly spread over some regions of the feature space. Hence by using the leaders alone it is not possible to find the probability density at a point whereas the counted leaders can be used to estimate the density at a point.

For a given threshold distance t, the counted-leader method works as follows. It maintains a set of leaders L, which is initially empty and is incrementally built. For each pattern x in D, if there is a leader l in L such that distance between x and l is less than t, then x

Figure 1. Semi spherical clusters



is assigned to the cluster represented by l and the count of the leader l is incremented. Note that, if there are many such leaders, then only one is chosen. If there is no such leader in L then x becomes a new leader which is added to the leaders set L and its count is initialized to one. The leaders and their respective counts depend on the order in which the data is scanned. But this does not affect the performance of the method which uses leaders because the cumulative count values of some of the neighboring leaders are used to estimate the density at a point. Further, when the data is drawn from a closed and bounded region of the feature space, the following theorem shows that the number of leaders is upper-bounded by a constant which is independent of the data set size and the distribution from which the data is drawn.

Theorem 1: Let the dataset D is drawn from a closed and bounded region S of the feature space. Then the number of leaders that can be derived from D using threshold distance t, such that $t > 0$, is at most $V_S / V_{t/2}$ where V_S is the volume of the region S and $V_{t/2}$ is the volume of a hyper-sphere of radius $t/2$.

Proof: Let the leaders set be L and $|L| = p$. For two distinct leaders l_1 and l_2 in L, it is guaranteed that distance between l_1 and l_2 is greater than or equal to t. Assume that at each leader l in L we replace a hyper-sphere of radius $t/2$. These hyper-spheres will not intersect each other. The total volume of these hyper-spheres is $pV_{t/2}$. Since, the data set is assumed to be drawn from a bounded region whose volume is V_S, we have $pV_{t/2} \leq V_S$. Hence $p \leq V_S / V_{t/2}$.

Counted K-Nearest Leader Classifier (ck-NLC)

The counted k-nearest leader classifier (ck-NLC) is similar to k-NNC where the selected prototypes replace the training set. A test pattern is classified based on majority cumulative count of a class among the k nearest leaders. It is described as follows. Let there are c classes *viz.*, $y^1,\ldots,y^c$. Let L_i be the prototypes derived using the *counted-leader* method using the training set for class y^i, for $i = 1$ to c. Let L be the set of all leaders. That is, $L = L_1 \cup \ldots \cup L_c$. For a given test pattern q, the k nearest leaders from L is obtained. For this k nearest leaders, the respective cumulative count for each class of leaders is obtained. Let this count for class y^i be W_i , for $i = 1$ to c. The classifier chooses the class label according to $argmax\{W_1, \ldots, W_c\}$. The method is described in Algorithm 1.

Algorithm 1 Counted k Nearest-Leader(L, q)
Step 1: Find the k-nearest leaders of q from L.
Step 2: Among the k-nearest leaders find the cumulative count of leaders that belongs to each class. Let this be W_i for class y^i, for i=1 to c.
Step 3: Find $W_x = argmax\{W_1, \ldots, W_c\}$.
Step 4: The class label assigned to q is y^x.

The domain from which a feature or attribute can take its values is almost always practically is a finite one, hence the assumption that the data set is drawn from a closed and bounded region of the feature space is not an unrealistic one. Under this assumption, the classification time of *ck-NLC* is *O(1)* only and the processing time of *l*-DBSCAN is *O(n)* only where n is the data set size. So the proposed methods are scalable. For the methods *k-NNC* and *ck-NLC*, the parameter k is found using a three fold cross validation from *{1, 2,…, 40}*.

An experimental study is done for one real time data set, *viz., Covtype.binary* available at the URL: *http://www.csie.ntu.edu.tw/cjlin/libsvmtools/datasets/binary.html* and one synthetic data set. *Covtype. binary* is a large data set consisting of 581012 patterns of 54 dimensions which belongs to two classes. The data is divided randomly into two parts consisting of 400000 and 181012 patterns, which are used as training and test sets, respectively. The synthetic data for a two dimensional two class problem is generated as follows. First class has 60000 patterns which are *i.i.d.* drawn from a normal distribution with mean as $(0,0)^T$ and covariance matrix as I_{2x2}(*i.e.*, identity matrix). Second class also is of 60000 patterns which is also *i.i.d.* drawn from a normal distribution with mean $(2.56,0)^T$ and covariance matrix as I_{2x2}. The Bayes error

rate for this synthetic data set is 10%. The data set is divided randomly into two parts consisting of 80000 and 40000 patterns which are used as training and testing sets respectively. The experimental results are summarized in Table 1 and Table 2 which shows that the proposed *ck-NLC* is a faster and efficient classifier for large data sets. Here the classifier *NLC* means the nearest leader classifier.

l-DBSCAN: A fast hybrid density based clustering method

A fast hybrid density based clustering method called *l*-DBSCAN is described below. The parameters used by *l*-DBSCAN are same as that of DBSCAN. That is, *l*-DBSCAN also uses the same ε, *MinPts* values as used by the DBSCAN method, *but l*-DBSCAN uses the set of leaders *L* and the leaders count values (which are derived by using the counted-leaders method) instead of the data set *D*. So the hybrid method primarily outputs clusters of leaders which can further be mapped into clusters of input data patterns by expanding (replacing) each leader by the pattern in *D* for which it is the representative. The key difference between the *l*-DBSCAN method and the DBSCAN method is in finding the number of points that are present in a hyper-sphere of radius ε at an arbitrary pattern of the data set. DBSCAN directly uses the data set and hence takes *O(n)* time where *n* is the data set size. But the *l*-DBSCAN method uses the set of leaders. If $l_1,\ldots,l_m$ are the leaders that are falling in the hyper-sphere, then the number of patterns in the hyper-sphere is taken to be $count(l_1) + \ldots + count(l_m)$ where count(*l*) gives the count for the leader *l*. If the total number of leaders is p, then finding $l_1,\ldots,l_m$ takes time equal to *O(p)*. Hence the total time requirement of the *l*-DBSCAN method is *O(np)*, whrereas that for the DBSCAN method is $O(n^2)$. The value of p is much smaller than n, hence *l*-DBSCAN is a faster method than DBSCAN.

Experiments are done to compare *l*-DBSCAN with DBSCAN. A *synthetic* and a standard data set are used for this purpose. The Synthetic data set called the banana data set is of a two dimensional one as shown in Figure 2 consisting of 4000 patterns. For this, the DBSCAN method with $\varepsilon = 16$ and *MinPts = 10* finds the two banana shaped clusters and also finds noisy outliers. The standard data set used is the *pendigits*

Table 1. Synthetic data set

Classifier	Classification Time(s)				Classification Accuracy(%)			
	Threshold				Threshold			
	0.25	0.20	0.15	0.10	0.25	0.20	0.15	0.10
NLC	880	1557	4137	11927	72.7	77.3	82.7	88.7
ck-NLC	1011	1744	4453	12154	74.7	78.4	84.0	90.5
NNC	1373066				94.89			
k-NNC	1373066				94.89			

Table 2. Covtype data set

Classifier	Classification Time(s)				Classification Accuracy(%)			
	Threshold				Threshold			
	0.05	0.04	0.03	0.02	0.05	0.04	0.03	0.02
NLC	54	74	105	151	75.52	77.35	79.80	82.43
ck-NLC	240	323	445	714	89.56	89.57	89.60	89.50
NNC	4692				85.17			
k-NNC	7884				89.61			

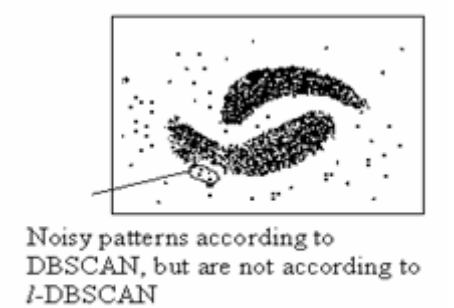
Table 3. Experimental results

Data Sets	Leaders Threshold	Rand-index	DBSCAN's execution time (s)	*l*-DBSCAN's execution time (s)
Synthetic	8	0.981	26.42	0.079
	4	0.993		0.237
	2	1		0.711
Pendigits	25	0.982923	696.86	7.58
	20	0.997139		15.33
	15	0.999988		27.52

data from the UCI machine learning repository (URL: *http://www.ics.uci.edu/mlearn/MLRepository.html*). The total number of patterns (obtained by combining the training and test data sets) is 10992. The number of dimensions is 16. The ε value used is 40 and *MinPts* is 23.

Rand-index (Rand, 1971) is used to compare the clustering outputs obtained by the DBSCAN method and the *l*-DBSCAN method. Note that, if rand-index = 1, then both clustering results are same. For the leaders threshold value 2, for the synthetic data set the rand-index obtained is 1. When the leaders threshold is 15, for the pendigits data set, the rand-index obtained is 0.999988. But the execution time of *l*-DBSCAN is less than 3% for the synthetic data set and less than 4% for the Pendigits data set when compared with that of the DBSCAN method. The results are summarized in Table 3. Figure 2 presents a situation where the leaders threshold is 8 and the clustering result of the *l*-DBSCAN is different from that of the DBSCAN.

Figure 2. The banana data set



FUTURE TRENDS

Building an index over the prototypes can reduce the computational time further. An indexing method like R-tree (Guttman, 1984) can be used for this purpose.

CONCLUSION

Non-parametric classification and clustering methods are popular and it performs very well with large data sets, but has huge computational burden. Using only a few selected prototypes which are rich enough to compute the probability density can reduce the computational time requirement. The chapter proposed to use a modified leaders clustering method where along with the leaders, the number of patterns grouped with the leader called the leader's count are preserved. The leaders set is used instead of the given data set with a nearest neighbor based classifier and with a density based clustering method. The classifier proposed is the counted *k* nearest leader classifier (ck-NLC) and the clustering method proposed is a faster version of the DBSCAN method called the *l*-DBSCAN method. The proposed methods are experimentally demonstrated to perform well and are shown to be scalable to work with large data sets like those in data mining applications.

REFERENCES

Asharaf, S., Murty, M.N., (2003). An adaptive rough fuzzy single pass algorithm for clustering large data sets, *Pattern Recognition* 36(12) 3015-18.

Cover, T., Hart, P. (1967): Nearest neighbor pattern classification. *IEEE Transactions on Information Threory* 13 21-27.

Dasarathy, B.V. (1991): Nearest neighbor(NN) norms: NN pattern classification techniques. *IEEE Computer Society Press*, Los Alamitos, California.

Duda,R.O, Hart,P.E & Stork,D.G. (2000). *Pattern Classification*, John Wiley Sons, 2nd Edition, Wiley-interscience Publication.

Ester,M., Kriegel,H.P., & Xu,X. (1996). A density based algorithm for discovering clusters in large spatial databases with noise, *In Proceedings of 2nd ACM SIGKDD*, Portand, Oregon, 226-231.

Guttman, A. (1984). R-trees: a dynamic index structure for spatial searching. *In Proceedings of the 13th ACM SIGMOD International Conference on Management of Data, Vol. 2. Boston, 47-57.*

Han,J., & Kamber, M. (2001). *Data Mining: Concepts and Techniques*, Morgan Kaufmann Publishers.

Hart. P.(1968): The condensed nearest-neighbor rule. *IEEE Transactions on Information Theory* IT-4 515-516.

Jain, A., Dubes, R., Chen, C.(1987). Bootstrap technique for error estimation. *IEEE Transaction on Pattern Analysis and Machine Intelligence* 9 628-633.

Jain, A.K, Murty, M.N. and Flynn, P.J. (1999): Pattern Clustering: A Review, *ACM Computing Surveys*, Sept. 1999, 264-323.

Rand, W.M, (1971). Objective criteria for the evaluation of clustering methods. *Journal of Classification*, Vol. 2, 193-218.

Spath,H. (1980). Cluster Analysis Algorithms for Data Reduction and Classification, Ellis Horwood, Chichester, UK.

Suresh Babu, V. & Viswanath, P. (2007) Weighted k-Nearest Leader Classifier for Large Data Sets, Proceedings of the Second *International Conference on Pattern Recognition and Machine Intelligence*, Indian Statistical Institute, Kolkata, 17-24.

Vijaya, P., Murthy, M. N., Subramanian, D.K.(2004): Leaders-subleaders: An efficient hierarchical clustering algorithm for large data sets. *Pattern Recognition Letters* 25. 505-513.

Viswanath,P. & Rajwala,P. (2006). A Fast Hybrid Density Based Clustering Method, *In Proceedings of the 18th Intl. Conf. on Pattern Recognition (ICPR-06)*, Hong Kong, IEEE Computer Society, Volume 1, 912-915.

S

KEY TERMS

Clustering of a dataset: The process of grouping data set into subsets whose union is equal to the dataset such that patterns in the same group are similar and the patternswhich are in different groups are dissimilar.

Cross Validation: It is a process of finding unknown parameters of a model by partitioning the training data set into several groups where each part in turn is used to test the model fitted by using the remaining parts.

Pattern: An object either physical or abstract which can be represented using a set of feature values. Normally a pattern is seen as a point in a feature space.

Partition of a Dataset: A collection of subsets of the dataset such that every pair of distinct subsets are disjoint and the union of the collection is equal to the dataset.

Prototype Selection: The process of selecting a few representative samples from the given training set suitable for the given task.

Rand-Index: If P and Q are two partitions of a data set, then *Rand-Index(P,Q)* which measures the similarity between P and Q is $(a+b)/d$, where a is the number of pairs (x,y) in the data set that x and y are present in a block of P and x and y are present in a block of Q, b is the number of pairs (x,y) in the data set such that x and y are not present in a block according to P and also not present in a block according to Q, and d is the total number of pairs present in the data set.

Training Set: The set of patterns whose class labels are known and which is used by the classifier in classifying a given pattern.

Scientific Web Intelligence

Mike Thelwall
University of Wolverhampton, UK

INTRODUCTION

Scientific Web Intelligence (SWI) is a research field that combines techniques from data mining, web intelligence and scientometrics to extract useful information from the links and text of academic-related web pages, using various clustering, visualization and counting techniques. Its origins lie in previous scientometric research into mining offline academic data sources such as journal citation databases, in contrast to Web Science, which focuses on engineering an effective Web (Berners-Lee et al., 2006). Typical scientometric objectives are either evaluative: assessing the impact of research; or relational: identifying patterns of communication within and between research fields. From scientometrics, SWI also inherits a need to validate its methods and results so that the methods can be justified to end-users and the causes of the results can be found and explained.

BACKGROUND

The term 'scientific' in Scientific Web Intelligence has a dual meaning. The first meaning refers to the scope of the data: it must be academic-related. For example, the data may be extracted from university web sites, electronic journal sites, or just pages that mention or link to academic pages. The second meaning of 'scientific' alludes to the need for SWI research to use scientifically defensible techniques to obtain its results. This is particularly important when results are used for any kind of evaluation.

Scientific Web Intelligence is young enough that its basic techniques are not yet established (Thelwall, 2005c). The current emphasis is on methods rather than outputs and objectives. Methods are discussed in the next section. The ultimate objectives of typical developed SWI studies of the future can be predicted, however, from research fields that have used offline academic document databases for data mining purposes. These fields include bibliometrics, the study of academic documents, and scientometrics, the measurement of aspects of science, including through its documents (Borgman & Furner, 2002).

Evaluative scientometrics develops and applies quantitative techniques to assess aspects of the value of academic research or researchers. An example is the Journal Impact Factors (JIF) of the Institute for Scientific Information (ISI) that are reported in the ISI's journal citation reports. JIFs are calculated for journals by counting citations to articles in the journal over a fixed period of time and dividing by the number of articles published in that time. Assuming that a citation to an article is an indicator of impact (because other published research has used the article in order to cite it), the JIF assesses the average impact of articles in the journal. By extension, 'good' journals should have a higher impact (Garfield, 1979), so JIFs could be used to rank or compare journals. In fact the above argument is highly simplistic. Scientometricians, whilst accepting the principle of citations as a useful impact proxy, will argue for more careful counting methods (e.g., not comparing citation counts between disciplines), and a much lower level of confidence in the results (e.g., taking them as indicative rather than definitive) (van Raan, 2000). Evaluative techniques are also commonly used for academic departments. For example, a government may use citation-based statistics in combination with peer review to conduct a comparative evaluation of all of the nation's departments within a given discipline (van Raan, 2000). Scientific Web Intelligence may also be used in an evaluative role, but since its data source is only web pages, which are not the primary outputs of most scientific research, it is unlikely to ever be used to evaluate academics' web publishing impact. Given the importance of the web in disseminating research (e.g., Lawrence, 2001), it is reasonable to measure web publishing, however.

Relational scientometrics seeks to identify patterns in research communication. Depending upon the scale of the study, this could mean patterns of interconnections of researchers within a single field, of fields or journals within a discipline, or of disciplines within the whole of science. Typical outputs are graphs of the relationships, although dimension-reducing statistics such as factor analysis are also used. For example, an investigation into how authors within a field cite each other may yield an author-based picture of the field that usefully identifies sub-specialisms, their main actors and interrelationships (Lin, White, & Buzydlowski, 2003). *Knowledge domain visualisation* (Börner, Chen, & Boyack, 2003) is a closely related research area, but one that focuses on the design of visualisations to display relationships in knowledge domains. Relationship identification is likely to be a common outcome for future SWI applications. An advantage of the web over academic journal databases is that it can contain more up to date information, which could help produce more current domain visualisations. The disadvantage, however, is that the web contains a wide variety of information that is loosely related to scholarly activity, if at all, even in university web sites. The challenge of SWI, and the rationale for the adoption of web intelligence and data mining, is to extract useful patterns from this mass of mainly useless data. Successful SWI will be able to provide early warning of new research trends, within and between disciplines.

MAIN THRUST OF THE CHAPTER

Scientific Web Intelligence uses methods based upon web links (web structure mining) and text (web content mining). A range of relevant web content and structure mining techniques are described below.

Academic Web Structure Mining

Modelling

Early academic web structure mining sought to assess whether counts of links to university or department web sites could be used to measure their online impact. This originated in the work of Ingwersen (1998). In brief, the results of this line of research indicated that links between university web sites, unlike citations, almost never represented knowledge transfer within the context of research. For example, few of these links point to online journal or conference articles. Nevertheless, it seems that about 90% of links are related in some way to academic activities (Wilkinson et al., 2003), and counts of links to universities correlate significantly with measures of research productivity for universities (Thelwall & Harries, 2004) and departments in some disciplines (Li et al., 2003; Tang & Thelwall, 2003). These results are consistent with web publishing being a natural by-product of research activity: people who do more research tend to create more web pages, but the chances of any given web page being linked to does not depend upon the research capabilities of its author, on average. In other words, more productive researchers tend to attract more links, but they also tend to produce more content and so the two factors cancel out (see also Barjak & Thelwall, 2007).

A little more basic information is known about academic web linking. Links are related to geography: closer universities tend to interlink more (Thelwall, 2002). Links are related to language: universities in countries sharing a common language tend to interlink more, at least in Europe, and English accounts for at least half of international linking pages in European universities in all countries except Greece (Thelwall, Tang & Price, 2003).

Data Cleansing

An important, but unexpected, outcome of the research described above was the need for extensive *data cleansing* in order to get better results from link counting exercises. This is because, on a theoretical level, link counting works best when each link is created independently by human experts exercising care and judgement. In practice, however, many links are created casually or by automated processes. For example, links within a web site are often for navigational purposes and do not represent a judgement of target page quality. Automatically generated links vary from the credit links inserted by web authoring software to links in navigation bars in web sites. The following types of link are normally excluded from academic link studies.

- All links between pages in the same site.
- All links originating in pages not created by the hosting organisation (e.g., mirror sites).

Note that the second type requires human judgements about ownership, and that these two options do not address the problem of automatically generated links. Some research has excluded a proportion of such links (e.g., Thelwall & Wilkinson, 2003) but an alternative more automated approach devised to solve this problem is to change the method of counting.

Several new methods of counting links have been devised. These are deployed under the umbrella term of Alternative Document Models (ADMs) and are, in effect, data cleansing techniques (Thelwall & Wilkinson, 2003). The ADMs were inspired by the realisation that automated links tended to originate in pages within the same directory. For example, a mini web site of 40 pages may have a web authorising software credit link on each page, but with all site pages residing in the same directory. The effect of these links can be reduced if links are counted between directories instead of between pages. In the example given, the 40 links from 40 pages would be counted as one link from a directory, discarding the other 39 links, which are now duplicates. The ADMs deployed so far include the page ADM (which is standard link counting) the directory ADM, the domain ADM and the whole site ADM. The choice of ADM depends partly upon the research question and partly on the data. A purely data-driven selection method has been developed (Thelwall, 2005a), designed to be part of a much more automated approach to data cleansing, namely Multiple Site Link Structure Analysis (MSLSA).

Subject Similarity and Clustering

A key SWI goal is to be able to automatically cluster academic web pages by academic subject. The ability to cluster web pages by (the more general concept of) topic has been investigated in the past, employing both text-based and link-based approaches. For example, the research of Chakrabarti et al. (2002) and Menczer (2005) shows that pages about the same topic tend to interlink more than with pages on different topics. It is logical to conclude that links will be helpful for subject clustering in academic webs.

A pair of web pages can be directly linked or may be indirectly connected by links if another page is joined to both by links. Direct links are not more reliable as indicators of subject than indirect connections, but indirect connections are far more numerous (Thelwall & Wilkinson, 2004). Hence, academic subject clustering should use both types.

There are many link-based clustering algorithms, but one that is fast and scalable is the Community Identification Algorithm (Flake et al., 2002). This accepts any number of interlinked pages as input and returns their community, based solely upon link structures. This 'community' is, loosely speaking, a collection of pages that tend to link to each other more than they link to pages outside of the community. Research with this algorithm on academic webs has shown that it is capable of identifying communities for the page, directory and domain ADM, but heavily linked pages negatively affect its results (Thelwall, 2003). Data cleansing to remove these pages is recommended.

Academic Web Content Mining

Academic web content mining is less developed than academic web structure mining, but is beginning to evolve. As with structure mining, a key goal is to be able to cluster academic web spaces by subject. There is some overlap between the two, for example in the need for ADMs and similar data cleansing.

Exploratory analysis of the text in university web sites has revealed the existence of many non-subject-specific high frequency words, such as computer and internet-related terms. Low frequency words were found to be predominantly not errors. The lesson for text mining is that low frequency words could not be ignored but that a strategy must be developed to filter out unwanted high frequency words (Thelwall, 2005b). Such a strategy, Vocabulary Spectral Analysis (VSA), has been developed (Thelwall, 2004). VSA is a technique based upon the standard vector space model and k-means clustering that identifies words that are highly influential in clustering document sets, and also words that are helpful for clustering document sets in ways that they do not naturally follow. This latter capability was developed in response to the realisation that academic web sites did not naturally cluster by subject, but in other ways, including university affiliation. Further research with low frequency words (Price & Thelwall, 2005) confirmed them to be helpful for subject clustering: removing them from the documents reduced their subject clustering tendency.

Knowledge Domain Visualisation

The field of Information Visualisation has been able to develop rapidly in recent years with the improved

speed and graphical power of PCs. Its newer subfield, *Knowledge Domain Visualisation* (KDViz) uses scientometric data and develops special purpose visualisations. These visualisations are for use by researchers to orient themselves within their own discipline, or to see how other fields or disciplines fit together or relate to each other. Although the typical data sources have been journal citation databases or journal text collections, these have similarities to web links and web content that make KDViz tools a logical starting point for Scientific Web Intelligence visualisations. A discussion of some KDViz research serves to highlight the visualisation capabilities already present.

- PNASLINK is a system that treats visualisations from articles published in the Proceedings of the National Academy of Sciences (White et al., 2004). It uses pathfinder networks (a technique for selecting the most important connections to draw for large network visualizations), and self-organising maps (a clustering technique that can plot documents on a two-dimensional map) to display information to users in order to help them select terms with which to search the digital library. Both text and citations are used by the algorithms.
- Cross maps is a technique for visualising overlapping relationships in journal article collections, (Morris & Yen, 2004). It produces two-dimensional graphs cross mapping authors and research fronts, more of a mainstream scientometrics application than PNASLINK.
- CITESPACE implements features that are designed to help users identify key moments in the evolution of research fields (Chen, 2004). It works by tracking the evolution of collections of papers in a field through citation relationships. Particularly important nodes in the generated network can be identified through the visualisations. Key moments in the time for the evolution of the network can also be revealed; these are called "turning points".

To apply all of the above visualisation techniques to SWI data is a future task, although some interesting progress has already been made (Heimeriks & van den Besselaar, 2006; Ortega, Aguillo, Cothey, & Scharnhorst, forthcoming). The main current challenge is to process web data in ways that make it possible to get useful results from visualisations.

FUTURE TRENDS

The immediate goal of SWI research is effective subject clustering of collections of academic web sites. This is likely to involve a fusion of link-based and text-based clustering approaches. Success will be dependent upon developing more effective data cleansing techniques. Perhaps initially these techniques will be only semi-automated and quite labour-intensive, but a longer-term goal will be to make them increasingly more automated. This prediction for a focus on data cleansing does not rule out the possibility that advanced web intelligence techniques could be developed that bypass the need for data cleansing.

The medium-term SWI goal is to harness academic web data to visualisations in order to give web information to users in a practical and effective way.

The long-term SWI goals are to develop applications that extend those of scientometrics and KDViz, to branch out into different web data sets, and to incorporate more web intelligence techniques (Zhong, Liu, & Yao, 2003) in order to extract new types of useful information from the data.

CONCLUSION

Scientific Web Intelligence has taken the first steps towards maturity as an independent field through the harnessing of techniques from scientometrics, web structure mining and web content mining. To these have been added additional techniques and knowledge specific to academic web spaces. Many of the new discoveries relate to data cleansing, recognition that web data is far 'noisier' than any data set previously used for similar purposes. The future is promising, however, particularly in the longer term if the techniques developed can be applied to new areas of web information – perhaps even to some that do not yet exist.

REFERENCES

Barjak, F., Li., X. & Thelwall, M. (2007). Which factors explain the web impact of scientists' personal homep-

ages? *Journal of the American Society for Information Science and Technology 58(2),* 200-211.

Berners-Lee, T., Hall, W., Hendler, J. A., O'Hara, K., Shadbolt, N., & Weitzner, D. J. (2006). A framework for Web science. *Foundations and Trends in Web Science*, 1(1), 1-130.

Borgman, C. & Furner, J. (2002). Scholarly communication and bibliometrics. In: Cronin, B. (ed.), *Annual Review of Information Science and Technology, 36*, Medford, NJ: Information Today Inc., 3-72.

Börner, K., Chen, C. & Boyack, K. (2003). Visualizing knowledge domains. Annual Review of Information Science & Technology, 37, 179-255.

Chakrabarti, S., Joshi, M.M., Punera, K. & Pennock, D.M. (2002). The structure of broad topics on the Web. *Proceedings of the WWW2002 Conference.* Available: http://www2002.org/CDROM/refereed/338/

Chen, C. (2004). Searching for intellectual turning points: Progressive knowledge domain visualization. Proceedings of the National Academy of Sciences, 101, 5303-5310.

Flake, G.W., Lawrence, S., Giles, C.L. & Coetzee, F.M. (2002). Self-organization and identification of web communities, IEEE Computer, 35, 66-71.

Garfield, E. (1979). Citation indexing: its theory and applications in science, technology and the humanities. New York: Wiley Interscience.

Heimeriks, G., & van den Besselaar, P. (2006). Analyzing hyperlink networks: The meaning of hyperlink-based indicators of knowledge. *Cybermetrics, 10*(1), Retrieved August 1, 2006 from: http://www.cindoc.csic.es/cybermetrics/articles/v2010i2001p2001.html.

Ingwersen, P. (1998). The calculation of Web Impact Factors. *Journal of Documentation*, 54(2), 236-243.

Lawrence, S. (2001). Free online availability substantially increases a paper's impact, *Nature*, 411(6837), 521.

Li, X., Thelwall, M., Musgrove, P. & Wilkinson, D. (2003). The relationship between the links/Web Impact Factors of computer science departments in UK and their RAE (Research Assessment Exercise) ranking in 2001, *Scientometrics*, 57(2), 239-255.

Lin, X., White, H.D. & Buzydlowski, J. (2003). Real-time author co-citation mapping for online searching. Information Processing & Management, 39(5), 689-706.

Menczer, F. (2005). Lexical and semantic clustering by web links. *Journal of the American Society for Information Science and Technology*, 55(14), 1261-1269.

Morris, S. & Yen, G. (2004). Crossmaps: Visualization of overlapping relationships in collections of journal papers. Proceedings of the National Academy of Sciences, 101, 5291-5296.

Ortega, J. L., Aguillo, I. F., Cothey, V., & Scharnhorst, A. (2008). Maps of the academic Web in the European higher education area - An exploration of visual Web indicators. *Scientometrics*, 74(2), 295-308.

Price, E.L. & Thelwall, M. (2005). The clustering power of low frequency words in academic webs. *Journal of the American Society for Information Science and Technology*, 56(8), 883-888

Tang, R. & Thelwall, M. (2003). Disciplinary differences in US academic departmental web site interlinking, *Library & Information Science Research*, 25(4), 437-458.

Thelwall, M., & Harries, G. (2004). Do better scholars' web publications have significantly higher online impact? *Journal of the American Society for Information Science and Technology*, 55(2), 149-159.

Thelwall, M., Tang, R. & Price, E. (2003). Linguistic patterns of academic Web use in Western Europe, *Scientometrics*, 56(3), 417-432.

Thelwall, M. & Wilkinson, D. (2003). Three target document range metrics for university Web sites. *Journal of the American Society for Information Science and Technology*, 54(6), 489-496.

Thelwall, M. & Wilkinson, D. (2004). Finding similar academic web sites with links, bibliometric couplings and colinks. *Information Processing & Management*, 40(1), 515-526

Thelwall, M. (2002). Evidence for the existence of geographic trends in university web site interlinking, *Journal of Documentation*, 58(5), 563-574.

Thelwall, M. (2003). A layered approach for investigating the topological structure of communities in the Web, *Journal of Documentation*, 59(4), 410-429.

Thelwall, M. (2004). Vocabulary Spectral Analysis as an exploratory tool for Scientific Web Intelligence. Proceedings of the 8th International Conference on Information Visualisation, 501-506.

Thelwall, M. (2005a). Data cleansing and validation for Multiple Site Link Structure Analysis. In: Scime, A. (Ed.), *Web Mining: Applications and Techniques*. Idea Group Inc., 208-227.

Thelwall, M. (2005b). Text characteristics of English language university Web sites. Journal of the American Society for Information Science and Technology, *56*(6), 609–619.

Thelwall, M. (2005c). Scientific Web Intelligence: Finding relationships in university webs. *Communications of the ACM*, 48(7), 93-96

van Raan, A.F.J. (2000). The Pandora's box of citation analysis: Measuring scientific excellence – the last evil? In: Cronin, B. and Atkins, H. B. (Eds.). The Web of knowledge: a festschrift in honor of Eugene Garfield. Metford, NJ: Information Today Inc. ASIS Monograph Series, 301-319.

White, H., Lin, X., Buzydlowski, J. & Chen, C. (2004). User-controlled mapping of significant literatures. Proceedings of the National Academy of Sciences, 101, 5297-5302;

Wilkinson, D., Harries, G., Thelwall, M. & Price, E. (2003). Motivations for academic Web site interlinking: Evidence for the Web as a novel source of information on informal scholarly communication, *Journal of Information Science*, 29(1), 59-66.

Zhong, N., Liu, J., & Yao, Y. (2003). *Web Intelligence*. Berlin: Springer-Verlag.

KEY TERMS

Alternative Document Model: A conceptual rule for grouping together web pages into larger units, such as sites and domains, for more effective data mining, particularly useful in web structure mining.

Knowledge Domain Visualisation: A subfield of Information Visualisation that is concerned with creating effective visualisations for specific knowledge domains.

Multiple Site Link Structure Analysis: A technique for identifying the alternative document model that best fits a collection of web pages.

Scientific Web Intelligence: A research field that combines techniques from data mining, web intelligence and scientometrics to extract useful information from the links and text of academic related web pages, principally concerning the impact of information and the relationships between different kinds of information.

Scientometrics: The quantitative study of science and scientists, particularly the documentary outputs of science.

Vocabulary Spectral Analysis: A technique using the vector space model and k-means clustering to identify words that are highly influential in clustering document sets.

Web Structure Mining: Data mining the web primarily through its link structure.

Web Content Mining: Data mining the web primarily through the contents of web pages, and ignoring interlinking between pages.

Seamless Structured Knowledge Acquisition

Päivikki Parpola
Helsinki University of Technology, Finland

INTRODUCTION

Some parts of this text, namely "Co-operative Building, Adaptation, and Evolution of Abstract Models of a KB" and most subsections in "Performing Reasoning in SOOKAT According to a KB", have appeared in an article (DOI:10.1007/s10115-004-0181-6) published in the 'Knowledge And Information Systems' journal (Parpola, 2004).

A knowledge base (KB) contains data and instructions for using it (e.g., as a rule base). A KB containing knowledge possessed by experts can be used in an expert system. It can solve problems requiring expert knowledge, explain its decisions and deal with uncertainty. An expert system can be used as a basis for a larger system, called a knowledge-based system (KBS).

Knowledge acquisition (KA) that is the development and maintenance of KBs, (e.g. an expert system), can be divided into several phases, performed sequentially and iteratively. Some phases may be performed in parallel with other phases. The most commonly recognised phases are requirements definition, analysis, design, and implementation.

Disintegration, or the gap between phases of development, especially between abstract and executable descriptions, was recognised during the early stages of KA (Marcus, 1988a; Motta, Rajan and Eisenstadt, 1988). It complicates the development of KBs and hinders traceability between parts of abstract and executable descriptions. •

BACKGROUND

Seamless Structured Knowledge Acquisition (SeSKA) (Parpola, 1998; Parpola, 1999a; Parpola, 1999b; Parpola, 2000) is a methodology for as well the development and maintenance of KBs as performing reasoning in them. It is designed to enhance integration of the KA process.

During KB construction, a series of models, including the (combined) dependency graph, the domain model, the inference model, together with analysis, design and implementation descriptions, is created and possibly modified. The structure of the knowledge base is based on the logical structure of the domain which has been noticed to be more stable than the component structure (Jacobson et al., 1992).

Related work concerning the use of metaobjects and metalevels in KA includes the following:

- Protégé-2000 (Fridman Noy, Ferguson and Musen, 2000) uses a metaobject protocol (Steele, 1990; Kiczales, des Riviers and Bobrow, 1991) to describe a model, for example, the CommonKADS model of expertise (Schreiber, Crubézy and Musen, 2000). This allows applications to be presented as instantiations of the model.
- OIL (Ontology Inference Language) (Fensel, van Harmelen, Decker, Erdmann and Klein, 2000) is a proposal, based on OKBC (Open Knowledge Base Connectivity), XOL (Ontology Exchange Language), and RDF (Resource Description Framework), for a joint standard for specifying and exchanging ontologies over the Internet. Modelling ontologies in OIL distinguishes three separate layers, the object level the first metalevel, and the second metalevel. The structure consists of several components. Rule bases, classes and slots, and types, as well as slot constraints and inheritance, are used. OIL is a frame-based system, using, for example, rule bases.

MAIN FOCUS OF THE CHAPTER

Models used in SeSKA

A domain model (DM) contains a somewhat stable componential structure of a domain. Knowledge is described through a network of relations between domain or abstract concepts with attributes. These attributes in the DM are selected according to what is needed in the dependency graph (DG).

Initial dependency graphs (DG) are acquired from different sources. DGs present inferential dependencies between attributes of DM concepts. Descriptions can be attached to dependencies. The actual DG is a combination of initial DGs. A DG contains dynamic knowledge described through a network of concept attributes and dependencies.

An inference structure (IS) presents the structure of possible inference sequences performed. The IS is shared among three sets of descriptions. Collections of analysis, design and implementation descriptions are attached to inferences in the IS. The result is called the inference model (IM).

These models can be described in terms of ontologies and natural language analysis (Parpola, 2000). The DM and the DG can be produced using several different KA techniques (Parpola, 1999b). Heterogeneous vocabulary can also be harmonised. The domain model and inference model can be instantiated to form the value model and execution model. This enables performing inferences.

Managing Change through Seamless Transformations

When constructing a KB with SeSKA, integration of a structured set of models can be produced through seamless transformations that is predefined ways of getting from objects in one model to objects in another model (Jacobson et al., 1992; Parpola, 1998). The KB structure is also maintained using the constructed shared skeleton: The inference structure (IS) describes the structure of possible inference sequences through a network of roles and inference steps. The former refer to concept attributes, and the latter have attached analysis, design, and implementation descriptions:

- The major logical components of abstract descriptions,
- Their formal descriptions, and
- Executable rules or functions, respectively.

The collections of different descriptions of all inference steps, in combination with the inference structure, form the analysis, design, and implementation models. The possibility of performing inferences, described in the models, requires instantiation of domain and inference models.

The idea of being able to describe a KB via models is proposed in the SeSKA methodology and implemented in the SOOKAT tool, described later.

The Knowledge Base Construction Process

Initial Formation of the Models Describing the Domain

The initial domain model and dependency graphs are formed on the basis of default value suggestions for, and dependency suggestions between, concept attributes, acquired from several knowledge sources that may give differing values.

Combining Dependencies and Attribute Values

Complementary dependency graphs can be processed using joining and simplification rules (Parpola, 1998). These rules allow different fragments of knowledge to be brought together, even before building a KB, and make it possible to show how they might be combined. Combination rules may accelerate the construction of a KB. To cope with contradictory or multiple attribute values, SeSKA defines combination heuristics (Parpola, 1999b).

Network of Roles and Inferences

A role in the IS is formed of concept attributes that a certain attribute depends on. Inferences between roles and descriptions associated with the inferences can be created on the basis of dependencies between attributes.

Several different dependency graphs can produce the same analysis model. One way to form such a dependency graph is to take the roles connected by an inference and set all the concepts referenced by the conclusion role to depend on all concepts referenced by the premise role. The analysis description of the inference can be attached to all dependencies.

Analysis descriptions are formalised to implementation descriptions, possibly via semi-formal design descriptions. The process is iterative and modular.

Managing Change through Seamless Transformations

Often, a need for change is acknowledged during the development or maintenance of a KB through imple-

mentation errors or other instabilities in the design or implementation models. The corresponding parts of the analysis model are traced using the shared inference structure. It may frequently be the practice to describe changes that have already been carried out, but it is important to keep the logical description up to date.

Possible Implementations of SeSKA

To implement SeSKA, an implementation paradigm should be able to define and modify entities with properties, relations between entities, inheritance, and instances. These facilities enable the presentation of metalevel constructions. At least the object-oriented (OO) approach, and conceptual graphs (CG) (Sowa, 1984) can be used. Either formalism is suitable for SeSKA, since they can be used for presenting both stable knowledge in the DM and dynamic knowledge in the DG, also supporting metalevel constructs. The tool SOOKAT has been built for testing different features of SeSKA.

Overview of the SOOKAT tool

A series of OO models is created and partially modified during the iterative development process.

Construction of a KB

SOOKAT uses two metaobject protocols, for domain and inference models, in order to be able to:

- Present and modify application instances and their attributes,
- At run time, create and modify concepts with instances, so that the modifications transfer to instances, and
- Use instantiations of the inference structure to apply abstract rules defined in the inference model to application instances of concept attributes.

Instantiation models are created in order to complete use of metaobject protocols. Taking full advantage of using the metaobject protocols means being able to simultaneously modify and use the knowledge base. The tool is implemented in the Java programming language, which does not inherently contain a metaobject protocol, unlike, for example, the Common Lisp programming language.

Architecture of SOOKAT

The class DomainModel contains an inheritance hierarchy of Concept metaclasses in the domain. The class ConceptAttribute can manage both the multiple values acquired from different knowledge sources and the default values worked out. A Concept contains instances of ConceptAttribute.

The class DependencyGraph refers to instances of the classes AttributeReference and Dependency. An AttributeReference contains a reference to a Concept in the domain model, as well as the name of a ConceptAttribute. A Dependency describes an inferential, (i.e., logical dependency between attributes of domain concepts). Dependencies of one or several types can be used. The type of a dependency is indicated by referring to an instance of a suitable subclass of the class DependencyType. A description is attached to each dependency.

Restricted views, showing only selected concepts, can be created for applications. This implements the Remove combination rule for DGs, without deleting information, as discussed earlier.

The model, managed through the class InferenceModel, is based on a network called the inference structure, which describes the structure of possible inferences through instances of the metaclasses Role and Inference. Inference defines, for each description level, a separate aggregate attribute:

- An analysis-level description is an abstract textual description. The initial analysis-level description of an inference is formed as a combination of descriptions of the dependencies the inference is based on. The description may be presented as a table.
- A design-level description is a semi-formal presentation of the analysis-level description. In SOOKAT, it is implemented as a rule table.
- An implementation-level description is composed of abstract descriptions of executable rules, implemented using the class Rule, containing three attributes, namely
  a. A premise expression that is, an instance of the class BooleanExpression, defining a logical operator, as well as operands that may be Expressions, RoleAttributeReferences, acting as variables, or arithmetic (integer) or logical (boolean) constants and references

to variables evaluating to primitive types,

b. A reference to the *conclusion attribute* that is, a RoleAttributeReference instance, and
c. A formula for obtaining the conclusion value that is an instance of the class ValueExpression which is a subclass of the class ArithmeticExpression. The conclusion attribute reference and the conclusion value form a BooleanExpression with an 'IS' operator.

In the value model, application instances are represented by instances of the Java class ConceptInstance. The values of their attributes are the possibly variable given values and the different possible conclusion values from the inferences. The latter depend on the execution model, in addition to the inference model.

The execution model contains instances of the Java classes RoleInstance and InferenceInstance. Messages are sent between these application instances in an order controlled by an instance of some subclass of the class ControlObject.

InferenceInstance instances adjust abstract rules. RoleAttributeReferences are replaced with corresponding values or references to attributes of application instances in the value model.

Co-Operative Building, Adaptation, and Evolution of Abstract Models of a KB

Suggestions for attribute values of concepts can be inserted in an arbitrary order by different knowledge sources. All suggestions are stored in instances of the class ConceptAttribute, a subclass of the class Attribute.

To eliminate heterogeneity in AttributeReference names, Concepts can be selected from among the ones in the domain model, as well as an attribute name from among those in a selected Concept. DependencyTypes can be selected from among the instances of it or its subclasses.

The combination of dependency graphs acquired from different sources is implicit in SOOKAT. Combination rules (Parpola, 1998) can be used incrementally, even when several knowledge sources are considered in parallel, assuming the context to be the same (Parpola, 2002; Parpola, 2001). Dependencies are joined automatically. Simplification is performed semi-automatically when SOOKAT collects dependencies. SOOKAT simply joins descriptions of varying dependencies and the user can remove duplicate descriptions. The descriptions and sources of the original dependencies are maintained. Descriptions of dependencies can also be augmented with lists of suitable contexts.

The dependency graph, with its descriptions, is used in forming the initial inference model, called the analysis model, consisting of the inference structure and abstract descriptions taken directly from dependencies. Inference structure formation is triggered from the user interface.

The informal, semi-informal, and formal descriptions are stored as attributes of instances of the class Inference, giving abstract descriptions of inferences in the inference model. Thus, formal and informal descriptions can be stored contiguously, as different descriptions can simultaneously be at different stages of development. Transformations between different descriptions of an inference are performed semi-automatically.

Different models can be modified in the user interface of SOOKAT, and models can be saved in a format that can be exported. Changes made can to some extent be propagated to other models (Parpola, 1999b).

Performing Reasoning in SOOKAT According to a KB

Instantiation of Models

In order to be able to perform inferences in an application, the domain model and inference model have to be instantiated to form the value model and execution model.

The Concepts in the domain model are gone through by the tool in order to remind the user of the items to be instantiated. For the desired concepts, the user can create one or more application instances with specific names and attribute values.

Roles in the inference model will be instantiated with selected collections of attribute instances. InferenceInstance instances between RoleInstance instances trigger rules of Inference metaobjects, applied to the ConceptInstance instance attributes, indicated by the RoleInstance instances.

A reason for defining Inference as a metaobject is that it provides a handy way of collecting together a group of InferenceInstances using the same rules. In this way, modifying a rule in a subclass of Inference affects all of its InferenceInstances at a time.

The Message Sending and Assignment Mechanism

Inferences are performed according to principles that have been adopted from the KADS methodology (Hesketh and Barrett, 1989; Schreiber, Akkermans, Anjewierden, de Hoog, Shadbolt, Van de velde and Wielinga, 1999) and modified. InferenceInstance instances between RoleInstance instances trigger rules of corresponding Inference instances, applied to the desired ConceptInstance attributes.

Assignments are made to values of Attributes of ConceptInstance instances in the value model, considered as variables to which either values of, or references to, the actual application instance attributes are assigned by InferenceInstance instances.

It has been proved that reasoning through the assignment of concept values holds the power of first-order logic (Wetter, 1990). Replacing concepts with attributes of concepts makes only a small addition to the proof.

The structure of possible inferences is defined by the instantiated inference structure.

Control Objects

A ControlObject (CO), specific to the chosen inference strategy, is used to control message sending between RoleInstance and InferenceInstance instances.

Actual inferencing takes place when COs for different inference strategies also guide the overall process as individual tasks. Message sending among CO, RoleInstance, and InferenceInstance instances (i.e. objects) are controlled by the CO. Suitable actions in RoleInstance and InferenceInstance objects are also triggered by the CO. The actual order of inferences to be performed is determined by the inference strategy used.

Using Protocols for Reasoning

Here a protocol means a documented series of reasoning stages that are gone through when a solution based on some premise values is reached (Ericsson and Simon, 1984). In other words, a protocol goes through an instantiation of one possible path (a sequence of inference steps, i.e., reasoning stages) through the inference structure. Reasoning stages are called protocol phases.

In SOOKAT, protocols are presented using dependencies. Each protocol phase depends on the attribute(s) reasoned during it, and on the previous phase. In other words, a DependencyGraph can describe the ordering of reasoning steps, in addition to logical dependencies between attributes.

The metaclass Protocol is defined as a subclass of the metaclass Concept. The class ProtocolPhase is, in turn, defined as a subclass of the class ConceptAttribute. As Protocol is a metaclass, the following are possible:

a. Dynamic creation of Protocol instances,
b. Inheritance among Protocol instances, and
c. Instantiation of Protocol instances (e.g. MineralClassificationProtocol instances).

The use of the second or third possibilities could increase the flexibility of the system through enabling the association of different and alternative COs to instances of Protocols.

Inference Using PSMs

Inferences can also be performed according to some problem-solving methods (PSMs) such as cover-and-differentiate (Eshelman, 1988) or propose-and-revise (Marcus, 1988b; Leo, Sleeman and Tsinakos, 1994).

FUTURE TRENDS

In the SeSKA and SOOKAT domain in the future, control objects (COs) for different problem-solving methods (PSMs) will be implemented, and their use investigated.

Mechanisms for transferring metaobjects, objects and their properties, as well as descriptions of PSMs and COs over the Internet between applications should be integrated in SOOKAT. OIL (Ontology Inference Language) mentioned in the section *BACKGROUND* is an interesting tool for this. Using Semantic Web to connect to and even discover knowledge in services or other applications is also a very interesting possibility (Hepp, 2005; Lopez, Sabou and Motta, 2006; Sabou and Pan, 2007).

CONCLUSION

The SeSKA (seamless structured knowledge acquisition) methodology has been developed to reduce

disintegration in the knowledge acquisition (KA) process. SeSKA does this by using *uniform formalisms* in the models used in KA phases, using automatic and semi-automatic *seamless transformations* between the models, facilitating traceability by linking them with different types of inference descriptions in the common inference structure (IS), inferencing in the models, using metaobject protocols to simultaneously modify and use knowledge bases (KB). Parts of the above means have been used in existing knowledge acquisition (KA) tools to overcome the disintegration problem, whereas SeSKA combines them.

REFERENCES

Ericsson, K. and Simon, H. (1984). *Protocol analysis*. MIT Press, Cambridge, MA, USA.

Fensel, D., van Harmelen, F., Decker, S., Erdmann, M., and Klein, M. (2000, October 2-6). OIL in a nutshell. In Dieng, R. and Corby, O., (eds.). *Knowledge Engineering and Knowledge Management. Methods, Models and Tools. 12th International Conference, EKAW 2000*. Juan-les-Pins, France.

Fridman Noy, N., Ferguson, R. W., and Musen, M. A. (2000, October 2-6). The knowledge model of Protégé-2000: Combining interoperability and flexibity. In Dieng, R. and Corby, O., (eds.). *Knowledge Engineering and Knowledge Management. Methods, Models and Tools. 12th International Conference, EKAW 2000*. Juan-les-Pins, France. Springer-Verlag.

Hepp, M. (2005). Products and services ontologies: A methodology for deriving ontologies from industrial categorization standards. *International Journal on Semantic Web and Information Systems, 2*(1).

Hesketh, P. and Barrett, T., editors (1989). *An Introduction to the KADS methodology*. STC Technology Ltd., Harlow, UK. ESPRIT Project P1098, Deliverable M1.

Jacobson, I., Christerson, M., Jonsson, P., and Övergaard, G. (1992). *Object-oriented software engineering, A use case driven approac*h. Addison-Wesley, Reading, Massachusetts, USA.

Kiczales, G., des Riviers, J., and Bobrow, D. (1991). *The art of the metaobject protocol.* MIT Press, Cambridge, MA, USA.

Leo, P., Sleeman, D., and Tsinakos, A. (1994). S-SALT, a problem solver plus; knowledge acquisition tool which additionally can refine its knowledge base. In *Proceedings of EKAW-94, the 8th European Knowledge Acquisition Workshop. Hoegaarden, Belgium, Artificial Intelligence Laboratory of the Vrije Universiteit Brussel*. Retrieved from http://arti.vub.ac.be/ekaw/welcome.html.

Lopez, V., Sabou, M., Motta, E. (2006) Mapping the real Semantic Web on the fly. *International Semantic Web Conference*. Georgia, Atlanta.

Marcus, S., editor (1988a). Automating knowledge acquisition for expert systems. *Kluwer International Series in Engineering and Computer Science*. Kluwer Academic Publishers, Boston.

Marcus, S. (1988b). *Salt: A knowledge-acquisition tool for propose-and-revise system*s. In (Marcus, 1988a), pp. 81-123.

Motta, E., Rajan, T., and Eisenstadt, M. (1988). A methodology and tool for knowledge acquisition in keats-2. In 3rd AAAI- *Sponsored Knowledge Acquisition for Knowledge-Based Systems Workshop*. Banff, Canada, 6-11 November, pages 21/1-20.

Parpola, P. (1998, April 18-23). Seamless development of structured knowledge bases. In B Gaines, M. M., (ed). *Proceedings of KAW98, Eleventh Workshop on Knowledge Acquisition, Modeling and Management. Banff, Alberta, Canada, , 1998. University of Calgary*. Retrieved http://ksi.cpsc.ucalgary.ca/KAW/KAW98/parpola/.

Parpola, P. (1999a, February 17-19). Development and inference in integrated OO models. In *Mohammadian, M., editor, CIMCA '99 - The international conference on computational intelligence for modelling, control and automation*. Vienna, Austria, . IOS Press.

Parpola, P. (1999b, May 26-29,). Applying SeSKA to Sisyphus III. In Fensel, D. and Studer, R., (eds). *Knowledge Acquisition, Modeling and Management. Proceedings of the 11th European Workshop, EKAW '99, Dagstuhl Castle, Germany, number 1621 in Lecture Notes in Artificial Intelligence*. Springer-Verlag.

Parpola, P. (2000, October 2-6). Managing terminology using statistical analysis, ontologies and a graphical KA tool. In Aussenac-Gilles, N., Biebow, B., and Szulman, S., (eds). In *Proceedings of the International*

Workshop on Ontologies and Texts during EKAW2000, 12th European Workshop on Knowledge Engineering and Knowledge Management, Juan-les-Pins, French Riviera. French group of interest TIA.

Parpola, P. (2001, August 4-10). Integration of development, maintenance and use of knowledge bases. In *Proceedings of the Workshop on Knowledge Management and Organizational Memory during IJCAI '01, International Joint Conference on Artificial Intelligence,* Seattle, Washington, USA. IJCAI consortium.

Parpola, P. (2002). Integration of development, maintenance and use of knowledge bases (revised version). In Dieng, R. and Matta, N., (eds.). *Knowledge Management and Organizational Memory,* pp. 41-50. Kluwer Academic Publishers.

Parpola, P. (2004). Inference in the SOOKAT object-oriented knowledge acquisition tool. *Knowledge and Information Systems, 8,* 310-329.

Sabou, M., and Pan, J. (2007). Web semantics: Science, services and agents on the World Wide Web. *Journal of Web Semantics*. Retrieved from DOI:10.1016/j.websem.2006.11.004. Elsevier.

Schreiber, A., Akkermans, J. M., Anjewierden, A. A., de Hoog, R., Shadbolt, N. R., Van de velde, W., and Wielinga, B. (1999). *Knowledge Enginering and Management: The CommonKADS Methodology. MIT Press.* Cambridge, MA.

Schreiber, G., CrubÈzy, M., and Musen, M. (2000). A case-study in using ProtÈgÈ-2000 as a tool for CommonKADS. In Dieng, R., and Corby, O., (eds), *EKAW2000, 12th European Workshop on Knowledge Engineering and Knowledge Management, Juan-les-Pins, French Riviera, October 2-6*. Springer-Verlag.

Sowa, J. (1984). *Conceptual structures: Information processing in mind and machine*. Addison-Wesley, Reading, Massachusets.

Steele, G. L. (1990). Common *lisp, the language, second editio*n. Digital Press, USA.

Wetter, T. (1990). First-order logic foundations of the KADS conceptual model. In Wielinga, B. et al. (eds.) *Current Trends in Knowledge Acquisition*.

KEY TERMS

Knowledge Base: Combination of both knowledge (facts about the world) and instructions of using the knowledge (inference, i.e., reasoning rules).

Knowledge Acquisition: Acquiring from possibly multiple experts, using multiple acquisition techniques, values of items of a knowledge base.

Natural Language Analysis: Analysing language (or signs presenting it), used (produced) by human beings, with special methods.

Object-Oriented: A way of describing the world or a part of it through active entities, called objects, that can be specialisations or parts of other objects.

Reasoning: Obtaining values of concept attributes, marked to be conclusion attributes, based on performing reasoning on concept attributes, marked to be premises.

Metaobject Protocol: Contains objects on three levels of abstraction:

- A Metaclass is a class, the instances of which are themselves classes.
- Class attributes with types describe a group of objects. Behaviour is defined through methods. Classes form an inheritance hierarchy.
- An Instance is an instance of another class than a metaclass.

SeSKA Methodology: The knowledge acquisition methodology SeSKA (seamless structured knowledge acquisition) has been developed to reduce disintegration between informal and formal descriptions of a knowledge base.

SOOKAT Tool: Implementation of the SeSKA methodology, used to test the methodology.

Search Engines and their Impact on Data Warehouses

Hadrian Peter
University of the West Indies, Barbados

Charles Greenidge
University of the West Indies, Barbados

INTRODUCTION

Over the past ten years or so *data warehousing* has emerged as a new technology in the database environment. "A data warehouse is a global repository that stores pre-processed queries on data which resides in multiple, possibly heterogeneous, operational or legacy sources" (Samtani et al, 2004).

Data warehousing as a specialized field is continuing to grow and mature. Despite the phenomenal upgrades in terms of data storage capability there has been a flood of new streams of data entering the warehouse. During the last decade there has been an increase from 1 terabyte to 100 terabyte and, soon to be 1 petabyte, environments. Therefore, the ability to search, mine and analyze data of such immense proportions remains a significant issue even as analytical capabilities increase.

The data warehouse is an environment which is readily tuned to maximize the efficiency of making useful decisions. However the advent of commercial uses of the Internet on a large scale has opened new possibilities for data capture and integration into the warehouse.

While most of the data necessary for a data warehouse originates from the organization's internal (operational) data sources, additional data is available externally that can add significant value to the data warehouse. One of the major reasons why organizations implement data warehousing is to make it easier, on a regular basis, to query and report data from multiple transaction processing systems and/or from external sources. One important source of this external data is the Internet.

A few researchers (Walters, 1997; Strand & Olsson, 2004; Strand & Wangler, 2004) have investigated the possibility of incorporating external data in data warehouses, however, there is little literature detailing research in which the Internet is the source of the external data. In (Peter & Greenidge, 2005) a high-level model, the Data Warehousing Search Engine (DWSE), was presented. However, in this article we examine in some detail the issues in search engine technology that make the Internet a plausible and reliable source for external data. As John Ladley (Ladley, 2005) states "There is a new generation of Data Warehousing on the horizon that reflects maturing technology and attitudes". Our long-term goal is to design this new generation Data Warehouse.

BACKGROUND

Data warehousing methodologies are concerned with the collection, organization and analysis of data taken from several heterogeneous sources, all aimed at augmenting end-user business function (Berson & Smith, 1997; Wixom & Watson, 2001; Inmon, 2003). Central to the use of heterogeneous data sources is the challenge to extract, clean and load data from a variety of operational sources. External data is key to business function and decision-making, and typically includes sources of information such as newspapers, magazines, trade publications, personal contacts and news releases.

In the case where external data is being used in addition to data taken from disparate operational sources, this external data will require a cleaning/merge/purge process to be applied to guarantee consistency (Higgins, 2003; Walters, 1997). The Web represents a large and growing source of external data but is notorious for the presence of bad, unauthorized or otherwise irregular data (Kim, 2003). Thus the need for cleaning and integrity checking activities increases when the web is being used to gather external data.

External data has the following advantages (Strand & Olsson, 2004) and disadvantages (Strand & Wangler, 2004):

Advantages

- Ability to compare internal data with external data
- Helps with the acquisition of data related customers
- Helps with the acquisition of additional data about the marketplace

Disadvantages

- Ensuring the quality of the external data
- Making users trust the external data
- Physically integrating the external data with the internal data
- Conceptually mapping the external data with the internal data

Given that the focus of this article is the use of external data (accessed from the web) in the data warehouse, it is in our interest to highlight the advantages and minimize or, if possible, eliminate the disadvantages mentioned above. We believe that one way of doing so is to use the appropriate search engine(s) to access the data from the Internet. Figure 1 illustrates the role played by external data in the decision-making process.

MAIN THRUST

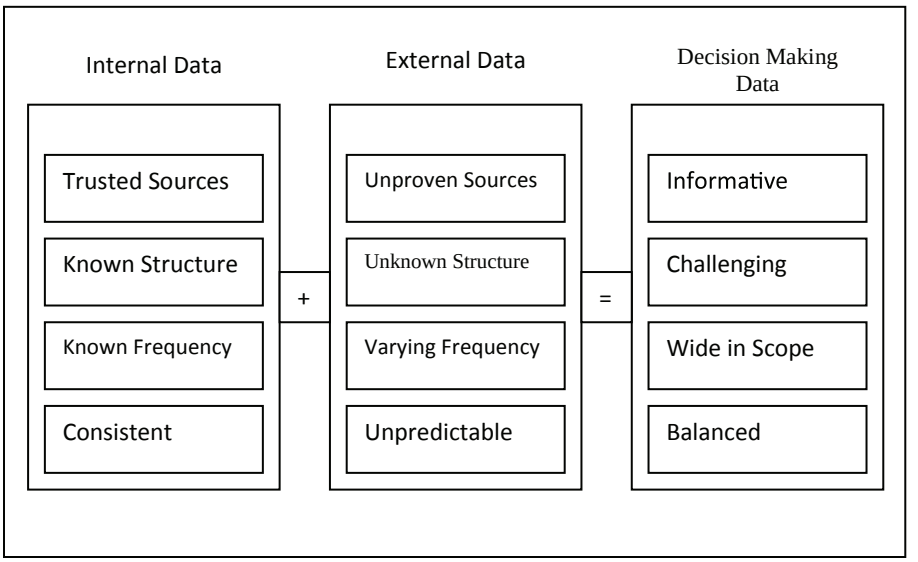
Search Engines: Comparison and Background Information

The literature is replete with examples of search engines for accessing external data from the Internet. In this section we examine a number of search engines and compare their characteristics.

There are substantial limitations to Search Engines. Most of them rely on crawler programs that go through a process of indexing web pages during a search (Butler, 2000). Most search engines only cover a fraction of the web, because vast amounts of data and documents stored in databases cannot be accessed by the current versions of web crawlers. These search engines are therefore unfortunately incapable of extracting information from the deep web or invisible web where much useful external data resides.

A long-standing shortcoming of search engines is that they provide only one way of organizing their search – by *salience (confidence, or certainty)* value (Bean, 2007). This value indicates how likely it is that a returned result matches the query. Such shortcomings

Figure 1. Merits of internal/external data



provide the motivation for a better approach to searching.

Another feature of conventional search engines is that they use algorithms and simple heuristics to rank web pages, and most of them only associate the text of a link with the page the link is on. However, newer search engines such as Google (see www.google.com) exploit the many links between web pages, and rank web pages based on *authorities* – web pages with many links pointing to them.

Given the obvious limitations of most search engines, and the sophistication of users, who are becoming more knowledgeable and demanding, the need for advanced search technologies is very urgent.

A number of research efforts have led to search engines with features that make them conducive to accessing external data. One approach is to use dedicated, specialized search engines. Examples are HotBot (Butler, 2000) which lets users search only academic sites with the domain name ending in ".edu", and NEC's ResearchIndex (now CiteSeer) (see http://citeseer.ist.psu.edu/citeseer.html) – a search engine designed specifically for scientists. Unlike most search engines Citeseer retrieves PDF and Postscript files. It uses simple rules based on the formatting of a document.

Advances in Artificial Intelligence (AI) technology promise better tools for web search. Techniques such as natural language, object recognition, and statistical machine learning are promising. Examples of new search engines that utilize AI techniques are MedStory, Powerset and Riya (Olsen, 2006). Powerset uses AI to train computers to make connections between words and make inferences. MedStory is a search engine focusing on health and is said to outdo Google, with a superior relevancy ranking. It uses a new approach to searching called *clustering*, and its strength lies "in the conceptual overview provided for each search topic and in the options provided for refining and focusing a search" (Anderson, 2007). The strength of Riya is that it is an "image" search engine. That is, it uses visual search to find images on the web. Details about how the search is performed can be accessed at http://www.ehow.com/how_2105171_visual-search-riya.html and http://www.techcrunch.com/2006/11/08/riyas-like-com-is-first-true-visual-image-search/.

Another class of search engines is designed for accessing data from the invisible or deep Web. Glenbrook Networks (2001) has developed a search engine that extracts data from multiple databases in the deep Web and then allows it to be searched from a single site. The method used is the analysis of a set of forms on a web page and then the use of AI techniques to sift through complicated web forms. Other examples of invisible Web search engines are BrightPlanet's Completeplanet (see www.Completeplanet.com) and Kartoo (see

http://websearch.about.com/od/enginesanddirectories/a/kartoo.htm

Another interesting search engine is Kenjin (Yurko, 2000) which uses neural networks and pattern matching technology. It reads and analyzes text on the screen, picks out major themes, and then searches the Internet for links related to those subjects.

The search technology, Clever (Butler, 2000), has taken the issue of relevancy to a new level. It enables search engines to automatically identify and display the most relevant sites it has crawled on the web. For most conventional search engines this can only be achieved by human intervention.

Inquirus (Lawrence & Giles, 1998) is a meta search engine which has been introduced to address some of the shortcomings of conventional search engines in searching the web. Meta search engines search the Web by making requests to multiple search engines. The major advantages of meta search engines are (a) ability to combine results of multiple search engines, and (b) ability to provide a consistent user interface for searching these search engines.

Desirable Features of Search Engine

The desirable features of a new search engine can be listed under the following headings:

(a) User-Interface design

- A better interface is probably a good thing for a new search engine – however, it has to be simple enough and give more information than any of the current solutions.
- Introduction of XML (Nassis, 2004), successor of HTML, for coding web pages should make it possible to restrict search

(b) Search box

- Most search engines provide only 2 levels of detail for users to input keywords – a simple text box and a formatted advanced search page with a few input fields.

A simple text box is great for most searches, however, the advanced search pages which are available do not allow users to adequately control input choices.

Suggestion: Have a more user-controlled advanced search page which allows users to:

(a) add or subtract search parameters and save these preferences, (b) have personalized search pages.
- Need for **semantic search** where search engine figures out what I mean from my keywords and can return results with the same meaning**.** It is frustrating and time-consuming to search the Internet for information and not be able to find it because of *semantics.*

(c) How results are presented
- Search is contextual – a small personal AI program running on the user's desk/web top should interface with the SE rather than natural language-based solutions.
- A search engine where one can see how changing the search criteria, changes the results on the fly.
- Develop a trend where users are focused on the quality of the information they receive and how reliable it is. Need for user-defined and AI-powered "TrustedSources" that can be used as a basis for ranking or filtering results.

(d) Relevancy
- Speed and relevance are required features of an efficient search engine.
- Need for "oracle capabilities" (or "magical mirror capabilities")
  - for example, if I am searching for "good restaurants for dinner", it would be good if I am given a list of potential restaurants next to where I live – this (intelligent) search should be based on my **IP address**. In other words the search engine should utilize *local context.*
- Automated technologies should help people to frame queries to maximize the relevance of search terms
- It would be useful if the search engine could answer with the most relevant website, instead of a webpage for user queries

(e) Specific versus general
- Highly personalized search portals that bring information one needs with regular updates. Also specialist search engines such as ones focused on, say, science or nature.
- Tackling a specific body of knowledge is more economical and efficient

Enhanced Active Search Engine (EASE)

The prevalence of so much data on the Internet means that it is potentially a superior source of external data for the data warehouse. Since such data typically originates from a variety of sources (sites) it has to undergo a merging and transformation process before it can be used in the data warehouse. The existence of the so called "invisible web", and ongoing efforts to gain access to these untapped sources, suggest that the future of external data retrieval will enjoy the same interest as that shown in internal data (Smith, 2001; Inmon, 2002; Sherman & Price, 2003).

The need for reliable and consistent external data provides the motivation for an intermediate layer between raw data gathered from the Internet (by search engines) and external data storage areas lying within the domain of the data warehouse (Agosta, 2000). This layer, which provides a data extraction/cleaning functionality, is called the *Meta-Data Engine* (M-D.E), and is a major component of the *Data Warehouse Search Engine* (D.W.S.E) Model (Peter & Greenidge, 2005). Figure 2 shows the structure of the D.W.S.E model.

The M-D.E enhances queries coming from the warehouse, and also captures, merges and formats information returned by the search engine (Lawrence & Giles, 1998; Peter & Greenidge, 2005). The M-D.E combined with the Search Engine component is called the *Enhanced Active Search Engine* (EASE).

EASE works as follows:

1. A query originates in the warehouse environment, and is modified by the M-D.E so that it is specific and free of "nonsense" words. A word that has a high occurrence in a text but conveys little specific information about the subject of the text is deemed to be a *nonsense* or *noise* word (Belew, 2000).

2. The modified query is transmitted to the search engine that performs its operations and retrieves its results documents.
3. The documents returned are analysed by the M-D.E and information is prepared for return to the warehouse.
4. The information relating to the answer to the query is returned to the warehouse environment.

The design of both search engine and data warehouse are skill-intensive and specialized tasks.

EASE: The Search Engine Component

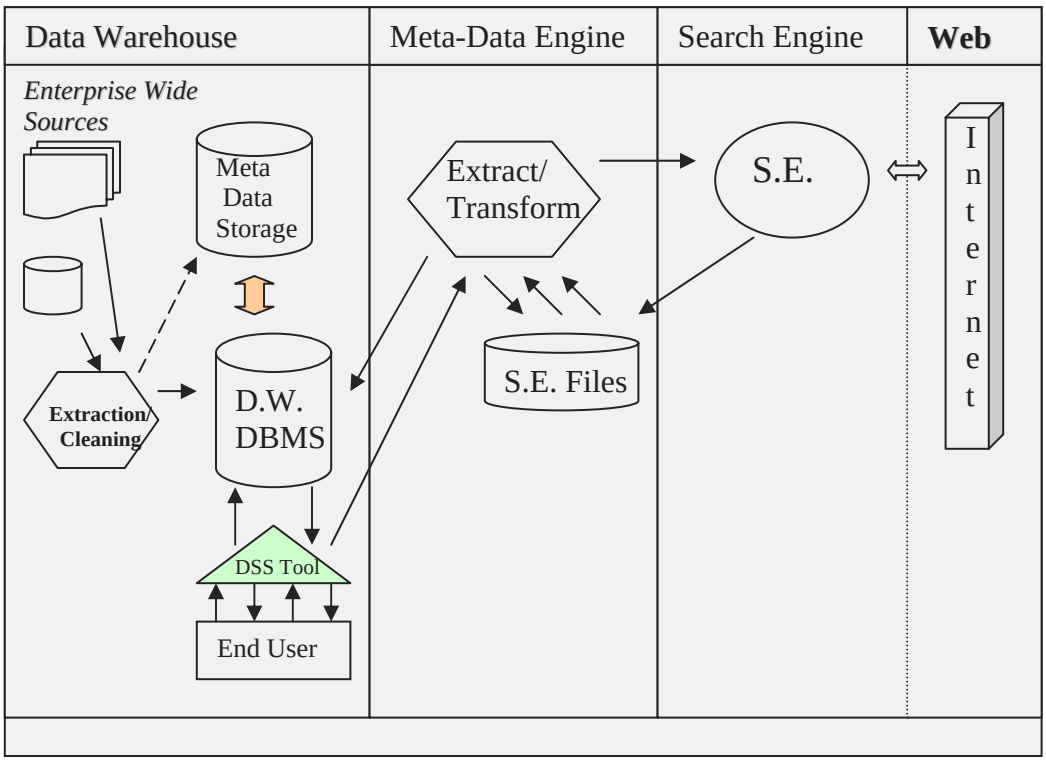
The search engine component of EASE is an important component, and has a major impact on the quality of the external data accessed from the Web. Accordingly, we are in the process of incorporating a number of features in that component – features we think will address some of the shortcomings of typical conventional search engines. Our search engine component is intended to have the following functionalities:

(a) The ability to conduct intelligent (semantic) searches using AI techniques (Olsen, 2006) such as natural language processing, machine learning, and neural networks (Yurko, 2000).
(b) The ability to access the invisible web.
(c) The ability to extract relevant data (Butler, 2000) from the Web.

EASE: The Meta-Data Engine Component

To cope with the radical differences between the search engine and data warehouse components we propose a Meta-Data Engine component to coordinate all activities. Typical commercial search engines are composed of a crawler (spider) and an indexer (mite). The indexer is used to codify results into a database for easy querying. Our approach reduces the complexity of the search engine by moving the indexing tasks to the meta-data engine. The meta-data engine seeks to form a bridge between the diverse search engine and data warehouse environments. The purpose of the meta-data engine is

Figure 2. The D.W.S.E. Model



to facilitate an *Automatic Information Retrieval* mode (see Figure 2).

The following are some of the main functions of the Meta-Data Engine

(a) Capture and transform data arising from the search engine - for example, handle HTML, XML, .pdf and other document formats
(b) Index words retrieved during search engine crawls
(c) Manage the scheduling and control of tasks arising from the operation of both DW and SE
(d) Provide a neutral platform so that SE and DW operational cycles and architectures cannot interfere with each other.

ANALYSIS OF MODEL

The model that we have provided in this article incorporates sophisticated search engine technology to ensure that external web data used to populate the data warehouse is of a high level of relevance. The following strengths of the model are identified:

- **Load sharing:** Since the meta-data engine component relieves the search engine component of tasks normally associated with information search engines, overloading is minimized.
- **Value added approach:** additional value is provided to the data warehouse, because the web data supplied by the sophisticated search engine EASE, is of a high level of relevance.
- The three-component architecture allows for a high degree of modularity, resulting in high levels of integrity and security in the external data.

A potential weakness of the model is that, with the rapid change in technologies, the two components of EASE have to be upgraded frequently.

FUTURE TRENDS

The changing nature of software makes it difficult to predict the future in the software industry. However, given the material provided in this article, the following are likely to be the relevant trends in the next few years in search engine and data warehouse technology:

- The use of ontologies in searching the Web (Corcho, 2006; Zhang et al, 2004).
- Design of more search engines to access the invisible web.
- Use of the technology in new search engines such as Swoogle (see http://swoogle.umbc.edu).
- Maturing of the Data Warehouse tools and techniques (Agosta, 2000; Inmon, 2002; Schwartz, 2003)
- The design and use of more dedicated and sophisticated search engines.
- Design the meta-data engine component to query multiple search engines (Lawrence & Giles, 1998) and, in so doing, speed up the search process.

We are confident that our D.W.S.E. model will gain prominence as practitioners realize the benefits of three distinct and architecturally independent layers.

CONCLUSION

In this article we presented a new approach to searching the (surface and deep) Web for relevant external data, which is subsequently integrated in the data warehouse. Our approach is based on a three-tiered model called the D.W.S.E model.

We believe that our model has the potential for harnessing the Web as a major source of information that can subsequently be used in the decision-making process. It is, however, important that we demonstrate the superiority of our model by comparing the results obtained using our approach with those obtained using the traditional external data approaches.

Although "relevance" remains a thorny issue, we have reason to believe that advances in Semantic Web developments – in particular, the use of ontologies – will help to address questions of relevance on the Internet.

REFERENCES

Anderson, P. A. (2007). "MedStory", J. Med. Libr Assoc. 95(2), pp 221-224.

Agosta, L. (2000). *The Essential Guide to Data Warehousing.* New Jersey: Prentice-Hall.

Bean, D. (2007). "How advances in Search Combine

Databases, Sentence Diagramming, and "Just the Facts". IT Pro, pp 14-19.

Belew, R.K. (2000). *Finding Out About: A Cognitive Perspective On Search Engine Technology and the WWW.* New York: Cambridge University Press.

Berson, A., & Smith, S.J. (1997). *Data Warehousing, Data Mining and Olap*. New York: McGraw-Hill.

Butler, D. (2000). "Souped-up Search Engines", Nature 405, pp. 112-115

Corcho, O (2006). "Ontology based document annotation: trends and open research problems", Int. J. Metadata Semantics and Ontologies, 1(1), 47-57.

Glenbrook Networks (2001) http://www.glendor.com/index.php?module=About&action=Index&tpl=cprimer

Hammersley, B. (2003). "Content Syndication with RSS", O'Reilly; 1 edition (256 pages)

Higgins, K.J. (2003). Warehouse Data Earns Its Keep. *Network Computing*, 14(8), 111-115.

Inmon, W.H. (2002). *Building the Data Warehouse, 3rd ed*. New York: John Wiley & Sons.

Inmon, W.H. (2003). The Story So Far. *Computerworld*, 37(15), 26-27.

Inmon, W.H. (2006). "How Do You Tune A Data Warehouse?". DM Review; 16(1). Available at http://dmreview.com

Kim, W., et al. (2003). "A Taxonomy of Dirty Data". *Data Mining and Knowledge Discovery*, 7, 81-99.

Ladley, J. (March 2007). "Beyond the Data Warehouse: A Fresh Look". DM Review Online. Available at http://dmreview.com

Lawrence, S. & Giles, C. L. (1998). "Inquirus, the NECI meta search engine, Computer Networks & ISDN Systems, 30(1), 95-105

Nassis, V. et al.(2004) "Conceptual Design of XML Document Warehouses", Y. Kambayashi, et al (Eds), DaWak 2004, LNCS 3181, pp 1-14.

Olsen, S. (2006). "Spying an intelligent Search Engine". CNET News.com.

Peter, H. & Greenidge, C. (2005) "Data Warehousing Search Engine". Encyclopedia of Data Warehousing and Mining, Vol. 1. J. Wang (ed), Idea Group Publishing, ISBN: 1591405572, pp. 328-333.

Ross, M. (Oct. 2006). "Four Fixes Refurbish Legacy Data Warehouses". Intelligent Enterprise, 9(10), 43-45. Available at http://www.intelligententerprise.com

Samtani, S. et al (2004). Recent Advances and Research Problems in Data Warehousing. Lecture Notes in Computer Science, Vol. 1552, Springer Berlin/Heildelberg, 81-92.

Schwartz, E. (2003). Data Warehouses Get Active. *InfoWorld*, 25(48), 12-13.

Shafi, S. M., & Rather, R. A. (2005). "Precision and Recall of Five Search Engines for Retrieval of Scholarly Information in the Field of Biotechnology." *Webology*, **2** (2), Article 12. Available at: http://www.webology.ir/2005/v2n2/a12.html

Sherman, C., & Price, G. (2003). The Invisible Web: Uncovering Sources Search Engines Can't See. *Library Trends*, 52(2), 282-299.

Smith, C.B. (2001). Getting to Know the Invisible Web. *Library Journal*, 126(11), 16-19.

Strand, M. & Olsson, M. (2003). The Hamlet Dilemma on External Data in Data Warehouses. *Proceedings of the 5th International Conference on Enterprise Information Systems (ICEIS)*, Angers, France, 570-573.

Strand, M. & Wangler, B. (June 2004). Incorporating External Data into Data Warehouses – Problem Identified and Contextualized. *Proceedings of the 7th International Conference on Information Fusion*, Stockholm, Sweden, 288-294.

Sullivan, D. (2000). Invisible Web Gets Deeper. *The Search Engine Report* [Electronic Version] retrieved January, 2001 from http://www.searchenginewatch.com.

Tan, X, Yen D.C., Fang, X. (Jan. 2003). "Web Warehousing: Web Technology meets Data Warehousing", Technology in Society, Vol. 25, No. 1, 131-148.

Walters, T. (March, 1997). Incorporating External Data into the Data Warehouse, *Proceedings of 22nd Annual*

SAS Users' Group International (SUGI) Conference, San Diego, California, 530-535.

Winter, R. & Burns, R. (2006). "Climb Every Warehouse". Intelligent Enterprise; 9(11) 31-35. Available at http://www.intelligententerprise.com

Wixom, B.H., & Watson, H.J. (2001). An Empirical Investigation of the Factors Affecting Data Warehousing Success. *MIS Quarterly*, 25(1), 17-39.

Yurko, C. (2000). "Search the Web without Searching", PC World.

Zhang, et al. (2004) "OntoSearch: An Ontology Search Engine". In *Proc. The Twenty-fourth SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence (AI-2004)*, Cambridge, UK.

KEY TERMS

Authority: A web page with many links pointing to it.

Data Integration: The process of combining data residing at different sources and providing the user with a unified view of these data.

Decision Support System (DSS): An interactive arrangement of computerized tools tailored to retrieve and display data regarding business problems and queries.

Deep (Invisible) Web : Denotes those significant portions of the web where data is stored which are inaccessible or cannot be readily indexed by the major search engines. The deep web represents an often ignored/neglected source of potential online information.

External Data: A broad term indicating data that is external to a particular company. Includes electronic and non-electronic formats.

Internal Data: Previously cleaned warehouse data which originated from the daily information processing systems of a company.

Information Retrieval: Denotes the attempt to match a set of related documents to a given query using semantic considerations. For example, Library catalogue systems often employ information retrieval techniques.

Metadata: Data about data; in the data warehouse it describes the contents of the data warehouse.

Search Situations and Transitions

Nils Pharo
Oslo University College, Norway

INTRODUCTION

Several studies of Web information searching (Agosto, 2002, Pharo & Järvelin, 2006, Prabha et al. 2007) have pointed out that searchers tend to satisfice. This means that, instead of planning for optimal search outcomes based on the best available knowledge, and on choosing the best information sources for their purpose, they aim at obtaining satisfactory results with a minimum of effort. Thus it is necessary to study other factors than the information needs and sources to explain Web search behaviour. Web information search processes are influenced by the interplay of factors at the micro-level and we need to understand how search process related factors such as the actions performed by the searcher on the system are influenced by various factors, e.g. those related to the searcher's work task, search task, knowledge about the work task or searching etc. The Search Situation Transition (SST) method schema provides a framework for such analysis.

BACKGROUND

Studies of information seeking and retrieval (IS&R) have identified many factors that influence the selection and use of sources for information seeking and retrieval. Web information searching often seems to be a rather haphazard behaviour where searchers seem to behave irrationally, i.e., they do not follow optimal textbook prescriptions (e.g., Ackermann & Hartman, 2003).

Other than the actual information need factors related to the searcher's personal characteristics, search task, work task, and social/organisational environment influence the searcher during his selection and use of information sources. These factors have been classified and discussed in great detail in the literature, and the SST method schema focuses specifically on the search process and how it is affected by external factors.

Early studies of Web searching to a large degree used log analysis (see review in Jansen and Pooch, 2001 and a summary in Spink and Jansen, 2004) or surveys (e.g., GVU's WWW user surveys (2001)) as their data collection methods. Log analysis can provide researchers with data on large numbers of user-system interactions focusing on users' actions. One common use has been to see how searchers formulate and reformulate queries (e.g., Spink et al, 2001). The user surveys have focused on demographics of web users and collected information on the use of different kinds of web resources, time spent on web use, e-shopping etc. Both these kinds of methods may reveal important information about how and why people use the Web, but they are unable to point out what causes the searcher to perform the actions he/she does. To learn how work tasks, search tasks, and searcher's personality directly affect Web information search processes the SST method schema (Pharo, 2002; Pharo & Järvelin, 2004) was developed.

MAIN THRUST OF THE CHAPTER

To present a method (e.g. Bunge, 1967), as well as a method schema (Eloranta, 1979), one needs to define its domain, procedure and justifications (Newell, 1969, Pharo, 2002). Both the domain and procedure is presented below to clarify the usability of the SST method schema.

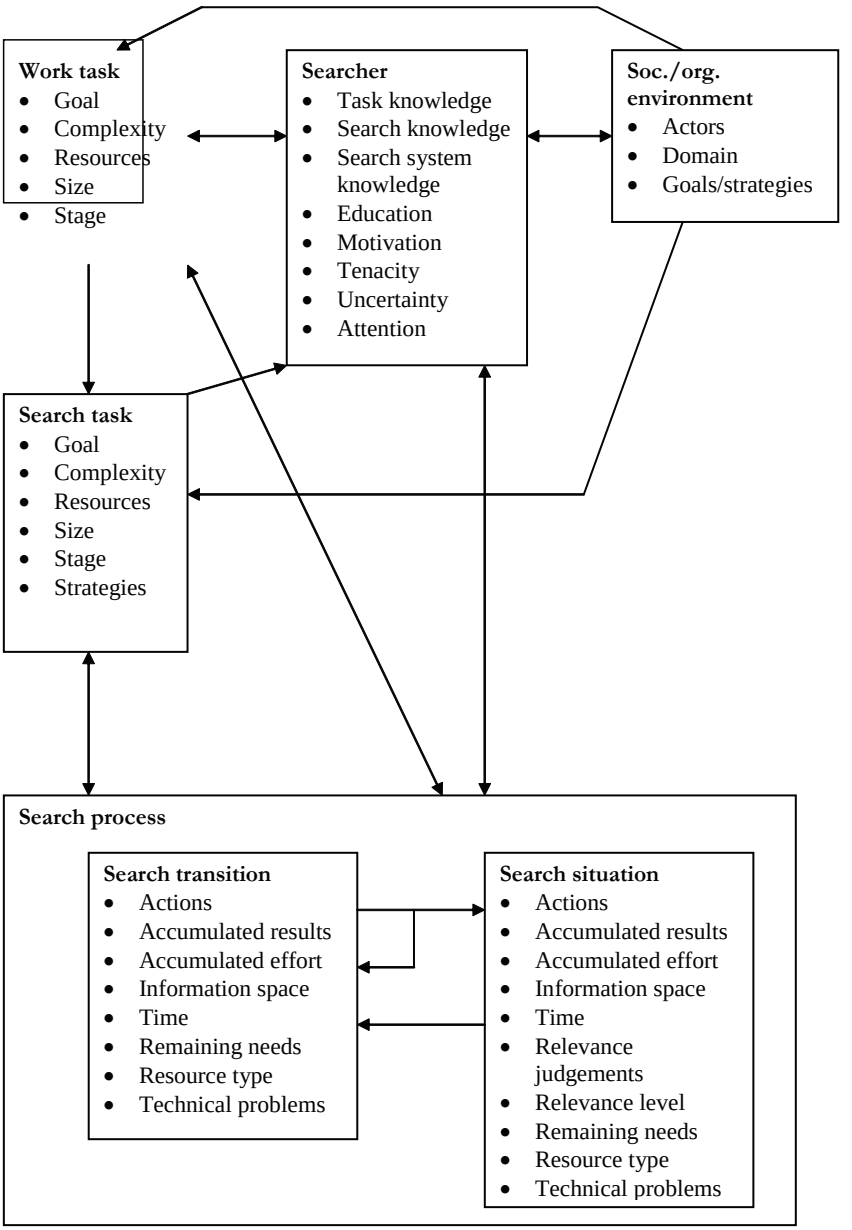
The Method Schema's Domain

The problem statement, or *domain*, which is used in the following, states the properties of the problem the method is intended for and their relationships. This designates how general it is possible to make the procedure for handling the problem.

Figure 1 is a representation of the framework's five *categories* and the relationships existing between them.

The *search process* category consists of two subcategories; *search situation* and *search transition*. The

Figure 1. The conceptual framework - the domain of the method schema



search process category will be emphasised here, the other categories and their attributes are well known from the IS&R literature (for details see Pharo, 2002).

Search *situations* are the periods during a search process when the searcher examines a resource in order to find information that may be of help in executing his work task. Situations may take place in the same kind of resources as transitions depending on the search task; if the searcher wants to learn more about the structuring of subject indices it would be natural to examine such resource types for that purpose.

Search *transitions* are executed in order to find resources in which the searcher believes there may be information that can help execute his task. The transitions

consist of source selection and inter-source navigation. An alternative way of explaining this is to say that while situations represent interaction with real information the transitions deal with *meta*-information.

Action is used to describe the moves made by the searcher during a situation/transition. In web interaction this includes the following of links, entering of queries, and reading of pages. The actions may be influenced, e.g., by a search task strategy.

The *accumulated results* refer to the information already found. This includes information found in previous situations as well as information found in the current one. Accumulated results relate to the completion of the information need (or the futility of trying this).

The *accumulated efforts* refer to how much work the searcher has had to invest from the start of the present session (or in prior sessions) up to the current position. In addition it can refer specifically to effort invested in the current situation.

The *information space* refers to the part of the Web that the searcher has navigated, as well as the information space anticipated by the searcher. The searcher has developed a cognitive model of the information space based on his knowledge about the Web and the existing resources on the Web, but also on his knowledge about institutions and organisations that he expects to be represented on the Web.

Time can be used to specify how the total amount of time spent during a search process influences the current situation, but it can also relate to the specific time used in that situation.

The *remaining needs* refer to what the searcher has planned to search for in the continuation of the search process and possibly in subsequent search processes.

Web *resource types* differ from each other with respect to content and format. Some are known from the world of paper-based publishing, such as newspapers, scientific journals, dissertations, novels, and collections of poems, but there are many new genres that have originated on the Web (home pages, blogs, wikis etc.) (Shepherd & Watters, 1998).

"Technical problems" is used to describe problems caused by the software in use, both on the client and server sides of interaction. Lack of bandwidth may also cause problems, for example in accessing resources that heavily depend on transmission of large amounts of data. Web pages that have disappeared also cause this kind of problem.

Situations and transitions share many attributes. Two unique attributes are only present in situations: relevance judgement and relevance level.

Relevance judgement relates to the searcher's evaluation of the pages found, which may be of use to him in different degrees. No predefined categories for relevance judgements are stated, in many studies binary (relevant or not relevant) or ternary (adding "partially relevant" to the former two) relevance measures have been used.

Relevance level signifies that the criteria used for evaluation may be related to the work task, which is what Saracevic (1996) calls situational relevance, but also be related to other levels, e.g., when an intermediary judges a resource's relevance for a (potential) user. Relevance judgements are also made in accordance with the organisational preferences, thus socio-cognitive relevance (Cosijn & Ingwersen, 2000) may also affect the judgements.

The Method Schema's Procedure

Log analysis and surveys are the most common data collection methods in Web IS&R. In general the problem with:

- Survey-type of web information searching (WIS) analysis is that neither the specific work tasks/ search tasks nor the specific processes are captured. Ex post facto findings in surveys provide only overviews of individuals' conceptions of WIS in general;
- Log analysis-type of data on WIS analysis is that it is not informed by anything happening in front of the computer screen.

In fact, even if one combines these types of analyses, one cannot analyse the processes properly for the effects of characteristics of work tasks, search tasks, or specific processes because the primary determinants are missing from the study setting. The use of triangulation as a general approach for data collection is necessary to capture the interplay of the various factors.

The procedure suggest the use of the following data collection methods for the domain described above:

- The search process can be captured using a multitude of methods such as transaction logs, screen captures from video logs, observation

and interviews. This kind of data will provide information on all of the proposed attributes of situations and transitions discussed above. It will also provide data on the other categories. Each attribute can be identified as follows:

- Actions are observable from transaction logs and screen captures.
- Accumulated results can be identified in the transaction logs and screen captures, but also through the use of pre- and post-search interviews.
- Accumulated efforts can be identified by searchers comments during the transactions and post-search interviews.
- Information space is dynamically changing during the search process and will be identified by analysing searchers comments, transaction logs and combining answers from pre- and post-search interviews.
- Time will be tracked in the transaction and video logs.
- Remaining needs can be identified through the interviews and the comments made by searchers during transactions.
- Resource types are easiest categorised from the video logs, but transaction logs can also be used.
- Technical problems can be observed using the video and transaction logs as well as the searchers' comments. Also post-search interviews should be addressed to collect this kind of information.
- Relevance judgements are identified by searchers bookmarking in transaction logs, but also from observation of their non-screen activities and post-search interviews. It is also possible to derive this from identifying what sources are listed in the references in, e.g., a report or thesis.
- Relevance level can be identified by the post-search-processing of the collected material. Are the sources used directly or indirectly in the searchers' output (reports etc)? The comments made by searchers during transactions and post-search interviews are also usable for identifying relevance level.

- The work task can be captured using a combination of interviews and output data, such as, e.g., theses, articles, reports and other written material.
- The search task can be identified from the interviews as well as utterances made by the searcher during the process (observation and video logs)
- The searcher can provide information about him/herself in interviews and questionnaires/surveys
- The social/organisational environment can be described through interviews, annual reports, and other written material documenting the organisation's activities and policies

The core data is usually collected using some kind of screen capturing or video recording of the computer screen during the processes. This, however, should be combined with simultaneous recordings of the searcher's utterances, and the searchers should be instructed to talk aloud (Ericsson & Simon, 1996) during searching. Alternatively Web transaction logs can be used alone, but then it would be difficult to capture non-action-related features of the process, e.g., to determine whether the searcher is actually reading a page or whether she is taking a coffee break.

The Method Schema's Justification

A method based on this schema was used to analyse real WIS interactions (Pharo, 2002) with encouraging results, in Pharo and Järvelin (2006) it was used to analyse the difference between textbook prescription of search strategies and searchers actual search strategies.

FUTURE TRENDS

The continuing growth, and hence importance, of the World Wide Web will make a better understanding of the complex interplay taking place during search processes even more important. The Web will doubtless be an important factor affecting interaction in various "environments" (business, education on all levels, research, public affairs etc). The need to analyse what takes place in different setting advocates the need for tools for holistic analysis, such as the SST method schema.

CONCLUSION

There is a large body of research literature on Web information searching (WIS). One approach to WIS research is log analysis, which is based on log contents and furnishes researchers with easily available massive data sets. However, the logs do not represent the user's intentions and interpretations. Another common approach to WIS is based on user surveys. Such surveys may cover issues like the demographics of users, frequencies of use, preferences, habits, hurdles to WIS etc. However, being ex post facto studies, they do not supply knowledge on how the searchers act in concrete WIS processes.

To understand and explain WIS processes, one needs to closely look at concrete processes in context. The literature of IS&R suggests several factors or categories like work task, search task, the searcher him/herself, and organisational environment as affecting information searching. A promising way to understand/explain WIS is through these categories. The current approaches to WIS, however, cannot shed light on what the effects are, if any. The SST method schema is developed to address these issues.

REFERENCES

Ackermann, E. & Hartman, K. (2003). *Searching and researching on the Internet and the World Wide Web*. 3rd ed. Wilsonville, Or.: Franklin, Beedle and Associates.

Agosto, D. E. (2002). Bounded rationality and satisficing in young people's Web-based decision making. *Journal of the American Society for Information Science*, 53 (1), 16-27.

Bunge, M. (1967). *Scientific research*. Heidelberg: Springer-Verlag.

Cosijn, E. & Ingwersen, P. (2000). Dimensions of relevance. *Information Processing & Management*, 36 (4), 533-550.

Eloranta, K. T. (1979). *Menetelmäeksperttiyden analyysi menetelmäkoulutuksen suunnittelun perustana* [The analysis of method expertise as a basis for planning education]. Tampere: Tampereen yliopiston.

Ericsson, K. A., & Simon, H. A. (1996). Protocol analysis: verbal reports as data. Cambridge, Mass: MIT Press.

GVU's WWW User Surveys (2001), GVU Center's WWW User Surveys. The Internet <http://www.cc.gatech.edu/gvu/user_surveys/>

Jansen, B. J. & Pooch, U. (2001). A review of Web searching studies and a framework for future research. *Journal of the American Society for Information Science*, 52 (3), 235-246.

Newell, A. (1969). Heuristic programming: Ill-structured problems. In Aronofsky, J. (ed.), *Progress in Operations Research, III*. New York: John Wiley & Sons, 360-414.

Pharo, N. (2002). *The SST method schema: a tool for analysing Web information search processes*. Tampere: University of Tampere [Doctoral dissertation].

Pharo, N. & Järvelin, K. (2004). The SST method: a tool for analysing Web information search processes. *Information Processing & Management*, 40 (4), 633-654.

Pharo, N. & Järvelin, K. (2006). 'Irrational' searchers and IR-rational researchers. *Journal of the American Society for Information Science and Technology*, 57 (2), 222-232.

Prabha, C., Connaway, L.S., Olszewski, L. & Jenkins, L.R. (2007). What is enough? Satisficing information needs. *Journal of Documentation*, 63 (1), 74-89.

Saracevic, T. (1996) Relevance reconsidered '96. In Ingwersen, P. and Pors, N.O., (Eds.) *Information Science: Integration in Perspective*. Copenhagen: Royal School of Librarianship, 201-218.

Shepherd, M. & Watters, C. (1998). The evolution of cybergenres. In *Proceedings of the 32nd Hawaii International Conference on System Sciences (HICSS '98)*, 97-109.

Spink, A., & Jansen, B. J. (2004). *Web Search: Public Searching of the Web*. Kluwer Academic Publishing.

Spink, A, Wolfram, D, Jansen, B. J. & Saracevic, T. (2001). Searching the Web: the public and their queries. *Journal of the American Society for Information Science*, 52 (3), 226-234.

KEY TERMS

Information System: A collection of sources containing potential information. Information systems can be of variable structure and size, from small bibliographic catalogues to the Web itself.

Method: A procedure for handling a set of problems. Methods can be categorised as "quantitative", which is, for example, the case for various statistical ways of data handling, or "qualitative", which may be exemplified by grounded theory. A method (and thus a method schema) consists of the following three parts: (1) a problem statement or *domain* modelling the phenomenon under study, (2) a *procedure* for collecting and analysing data to understand the phenomenon, and (3) a *justification*, e.g. by showing its ability to solve designated problems of the domain.

Method Schema: Any representation defined for one or more methods, where one or more aspects of the method have been left uninterpreted and represented only through their plain name, and where some aspects of the methods may have been left out (even lacking their naming). Method schemas take the format of a method, but it contains unspecified components that need to be specified if it is to reach the level of a method. In other words a method schema is an abstract representation of one or more methods – a generic model. The difference between a method and a method schema can be said to be a continuum of generality.

Search Process: The period during which a searcher interacts with an information system. The structure of a search process is dialectic; it switches between search situations and search transitions.

Search Situations: The periods of a search process during which the searcher interacts with sources potentially containing information related to his/her search task.

Search Transitions: The periods of a search process during which the searcher interacts with sources containing meta-information

SST Method Schema: A method schema developed to analyse search processes by identifying what external and internal factors interplay with the search process during, before, and after the process.

Secure Building Blocks for Data Privacy

Shuguo Han
Nanyang Technological University, Singapore

Wee Keong Ng
Nanyang Technological University, Singapore

INTRODUCTION

Rapid advances in automated data collection tools and data storage technology have led to the wide availability of huge amount of data. Data mining can extract useful and interesting rules or knowledge for decision making from large amount of data. In the modern world of business competition, collaboration between industries or companies is one form of alliance to maintain overall competitiveness. Two industries or companies may find that it is beneficial to collaborate in order to discover more useful and interesting patterns, rules or knowledge from their joint data collection, which they would not be able to derive otherwise. Due to privacy concerns, it is impossible for each party to share its own private data with one another if the data mining algorithms are not secure.

Therefore, privacy-preserving data mining (PPDM) was proposed to resolve the data privacy concerns while yielding the utility of distributed data sets (Agrawal & Srikant, 2000; Lindell.Y. & Pinkas, 2000). Conventional PPDM makes use of Secure Multi-party Computation (Yao, 1986) or randomization techniques to allow the participating parties to preserve their data privacy during the mining process. It has been widely acknowledged that algorithms based on secure multi-party computation are able to achieve complete accuracy, albeit at the expense of efficiency.

BACKGROUND

In recent years, PPDM has emerged as an active area of research in the data mining community. Several traditional data mining algorithms have been adapted to become privacy-preserving that include decision trees, association rule mining, *k*-means clustering, SVM, Naïve Bayes, and Bayesian network. These algorithms generally assume that the original data set has been horizontally and/or vertically partitioned with each partition privately held by one party. In privacy-preserving data mining algorithms, data are horizontally and/or vertically partitioned in order to spread the data across multiple parties so that no single party holds the overall data.

In this chapter, we focus on current work in privacy-preserving data mining algorithms that are based on Secure Multi-party Computation (Yao, 1986). In secure multi-party computation, there are two models that classify adversarial behaviors (Goldreich, 2002): the semi-honest model and the malicious model. Loosely speaking, a party in a semi-honest model follows the protocol properly but it keeps a record of all the intermediate computations during the execution. After the protocol, a party attempts to compute additional information about other honest parties. A party in the malicious model is allowed to diverge arbitrarily from the protocol. To force a malicious party to follow the protocol, zero-knowledge proofs can be applied. Zero-knowledge proofs as introduced by authors in (Goldwasser, Micali, & Rackoff, 1989) are proofs of the validity of an assertion made by a party without disclosing additional information.

MAIN FOCUS

In this section, we review current work on privacy-preserving data mining algorithms that are based on secure multi-party computation (Yao, 1986).

Privacy-Preserving Decision Trees

In (Lindell & Pinkas, 2000), the authors proposed a privacy-preserving ID3 algorithm based on cryptographic techniques for horizontally partitioned data involving two parties. The authors in (Du & Zhan, 2002) addressed the privacy-preserving decision tree

induction problem for vertically partitioned data based on the computation of secure scalar product involving two parties. The scalar product is securely computed using a semi-trusted commodity server. In the model, a semi-trusted third party helps two parties to compute scalar product; the third party will learn nothing about the parties' private data and is required not to collude with any of them. The authors in (Vaidya & Clifton, 2005a) extended the privacy-preserving ID3 algorithm for vertically partitioned data from two parties to multiple parties using the secure set intersection cardinality protocols.

Privacy-Preserving Association Rule Mining

In (Kantarcioglu & Clifton, 2004), the authors proposed a method to securely mine association rules for horizontally partitioned data involving three or more parties. The method incorporates cryptographic techniques to reduce the information disclosed. The authors in (Vaidya & Clifton, 2002) presented a privacy-preserving association rule mining algorithm for vertically partitioned data using secure scalar product protocol involving two parties. A secure scalar product protocol makes use of linear algebraic techniques to mask private vectors with random numbers. Solutions based on linear algebraic techniques are believed to scale better and perform faster than those based on cryptographic techniques. To extend association rule mining algorithm for vertically partitioned data to multiple parties, the authors in (Vaidya & Clifton, 2005b) proposed a secure set intersection cardinality protocol using cryptographic techniques. The communication and computation complexities of the protocol are O(mn) and $O(mn^2)$, where m and n are the length of private vectors and the number parties respectively.

Privacy-Preserving Clustering

The authors in (Vaidya & Clifton, 2003) presented a method to address privacy-preserving k-means clustering for vertically partitioned data involving multiple parties. Given a sample input that is partially held by different parties, determining which cluster the sample is closest must be done jointly and securely by all the parties involved. This is accomplished by a secure permutation algorithm (Du & Atallah, 2001) and a secure comparison algorithm based on the circuit evaluation protocol (Yao, 1986). The authors in (Geetha Jagannathan & Wright, 2005) proposed a new concept of arbitrarily partitioned data that is a generalization of horizontally and vertically partitioned data. They provided an efficient privacy preserving protocol for k-means clustering in an arbitrarily partitioned data setting. To compute the closest cluster for a given point securely, the protocol also makes use of secure scalar product protocols. The authors in (G. Jagannathan, Pillaipakkamnatt, & Wright, 2006) presented a simple I/O-efficient privacy-preserving k-clustering algorithm. They claimed that cluster centers produced by their algorithm are more accurate than those produced by the iterative k-means algorithm. The algorithm achieved privacy using secure scalar product protocols and Yao's circuit evaluation protocol (Yao, 1986). The authors in (Lin, Clifton, & Zhu, 2005) presented a technique that uses EM mixture modeling to perform clustering on horizontally partitioned distributed data securely. In the protocol, each partition is computed locally based on local data points. The global sum of the partitions from all parties is then computed without revealing the individual values by secure sum.

Privacy-Preserving Support Vector Machine

The authors in (Yu, Vaidya, & Jiang, 2006) proposed a privacy-preserving SVM classification algorithm for vertically partitioned data. To achieve complete security, the generic circuit evaluation technique developed for secure multiparty computation is applied. In another paper (Yu, Jiang, & Vaidya, 2006), the authors securely constructed the global SVM classification model using nonlinear kernels for horizontally partitioned data based on the secure set intersection cardinality protocol (Vaidya & Clifton, 2005b). The authors in (Laur, Lipmaa, & Mielikainen, 2006) proposed secure protocols to implement the Kernel Adaption and Kernel Perception learning algorithms based on cryptographic techniques without revealing the kernel and Gramm matrix of the data.

Privacy-Preserving Naïve Bayes

In (Kantarcioglu & Clifton, 2003), the authors presented a privacy-preserving Naïve Bayes classifier for horizontally partitioned data using secure sum—an instance the Secure Multi-party Computation (Yao, 1986). The

authors in (Vaidya & Clifton, 2004) developed privacy-preserving Naïve Bayes classifier for vertically partitioned data. Secure scalar products protocols are used to give the appropriate probability for the data while preserving data privacy. The same authors summarized the above two protocols for different partitioned data and published them in (Vaidya, Kantarcioglu, & Clifton, 2007). The authors in (Wright & Yang, 2004) presented a privacy-preserving Bayesian network computation for vertically distributed data involving two parties. They showed an efficient and privacy-preserving version of the *K2* algorithm to construct the structure of a Bayesian network for the parties' joint data based on secure scalar product protocols.

Privacy-Preserving Gradient Descent Paradigm

The authors in (Du, Han, & Chen, 2004) proposed secure versions of multivariate linear regression and multivariate classification for vertically partitioned data involving two parties. They developed a practical security model based on a number of building blocks, such as the secure matrix product protocol and the secure matrix inverse protocol. In (Barni, Orlandi, & Piva, 2006), the authors presented a privacy-preserving neural network involving two parities under the scenario where one party holds private data and the other party only has a private neural network that is used to process the private data. The protocol preserves the privacy of two parties using secure scalar product protocols, private polynomial evaluation, and so on. In (Wan, Ng, Han, & Lee, 2007), the authors proposed a generic formulation of gradient descent methods for secure computation by defining the target function as a composition. They demonstrated its feasibility in specific gradient descent methods, such as linear regression, and neural network. The privacy is preserved using secure scalar product protocols.

Other Recent Privacy-Preserving Data Mining Algorithms

In (Han & Ng, 2007a), the authors proposed a protocol for secure genetic algorithms for the following scenario: Two parties, each holding an arbitrarily partitioned data set, seek to perform genetic algorithms to discover a better set of rules without disclosing their own private data. The challenge for privacy-preserving genetic algorithms is to allow the two parties to securely and jointly evaluate the fitness value of each chromosome using each party's private data without compromising their data privacy. They proposed a new protocol to address this challenge that is correct and secure. The protocol preserves the privacy of two parties using secure scalar product protocol.

In (Han & Ng, 2007b), the authors proposed a protocol for privacy-preserving self-organizing map for vertically partitioned data involving two parties. Self-organizing map (SOM) is a widely used algorithm for transforming data sets to a lower dimensional space to facilitate visualization. The challenges in preserving data privacy in SOM are (1) to securely discover the inner neuron from data privately held by two parties; (2) to securely update weight vectors of neurons; and (3) to securely determine the termination status of SOM. The authors proposed protocols to address the above challenges using secure scalar product protocols. They proved that these protocols are correct and privacy-preserving. Also, they proved that the intermediate results generated by these protocols do not violate the data privacy of the participating parties.

FUTURE TRENDS

As shown in the above sections, much work has been done to equip conventional algorithms with the privacy-preserving feature. There are other directions of research in privacy-preserving data mining. One direction of inquiry asks the question: "What is the definition of privacy?" The authors in (Clifton, Kantarcioglu, & Vaidya, 2002) provided a definition for privacy and presented some metrics to evaluate the degree to which privacy is preserved. Most of the current work in privacy preserving data mining focuses on preserving the privacy of privately held data values from other parties. The definition and coverage of privacy also extends to the names of attributes, as the sharing of attribute names among multiple parties is a potential form of privacy violation. In order to achieve "perfect" security; i.e., to preserve the privacy of all possible private information, the definition of privacy for privacy-preserving data mining algorithms should be appropriately stretched. Is it possible for privacy-preserving data mining algorithms to fulfill the "perfect" security requirement?

Another direction is secure-satisfiability protocols: A protocol that is secure in one model (e.g., semi-honest model) may be insecure in another model (e.g., malicious model). So far the protocols proposed in the malicious model are more secure than the ones in the semi-honest model. However, as shown by the author (Goldreich, 2002), there are three problems that cannot be avoided even in the current malicious model:

- *Parties refusing to participate in the protocol (when the protocol is first invoked).* It is impossible to force the malicious parties to participate in the protocol.
- *Parties aborting the protocol prematurely (e.g., before sending their last message).* It is hard to prevent the malicious party from suspending or aborting the execution of the protocol. In particular, the malicious party may abort at the first moment when it obtains the desired result.
- *Parties substituting their local input (and entering the protocol with an input other than the one provided to them).* The malicious party may modify its local input to attack the private data of the honest parties. The protocol in the current malicious model cannot prevent the malicious party from giving the false data. But if the malicious party is allowed to give *arbitrary* false inputs, it is possible to probe the private inputs of the honest parties.

When the problems occur to the honest parties, it is an interesting research topic whether the privacy of the honest parties is violated. To address these problems, more researches are required to build a more secure model for data mining algorithms.

Third direction is whether what is done to preserve 100% of the privacy based on SMC is necessary or not. The most important reason is that the performance of data mining algorithms based on SMC is much worse than ones based on randomization technique. Sometimes, the performance may be more important for the parties. To achieve the high performance for privacy-preserving data mining algorithms while preserving the high privacy, the hybrid technique may be a solution. In the privacy-preserving data mining algorithms based on SMC, we use the randomization technique to preserve the privacy for less important data to increase the performance. But how to realize the idea in the specific data mining algorithms is another interesting research topic.

Finally most of the current privacy-preserving data mining algorithms focus on the algorithms in data mining domain. However, other algorithms in other domains, such as machine learning, computational intelligence, and so on, have been proposed to be privacy-preserving. For enabling the algorithms in these domains to be privacy-preserving, the main problems are (1) how to preserve the privacy between two for the iterative algorithms and (2) how to utilize the efficiency for the iterative privacy-preserving algorithms.

CONCLUSION

Since the first papers on privacy preserving data mining appeared in 2000, the data mining community has lent much attention to the area. Numerous extensions have been proposed to enable existing machine learning and data mining algorithms with the privacy-preserving feature. Work has also been done to enhance existing privacy-preserving data mining algorithms using the semi-honest model to securely work in the malicious model. This chapter presents the state-of-the-art in privacy-preserving data mining algorithms. The techniques covered include decision trees, association rule mining, clustering, naïve Bayes, support vector machines, and gradient descent algorithms.

REFERENCES

Agrawal, R., & Srikant, R. (2000). *Privacy-preserving data mining.* Paper presented at the Proceedings of the ACM international Conference on Management of Data, Dallas, TX, USA.

Barni, M., Orlandi, G., & Piva, A. (2006). *A privacy-preserving protocol for neural-network-based computation.* Paper presented at the Proceedings of the Multimedia and Security Workshop 2006, MM and Sec'06, Geneva, Switzerland.

Clifton, C., Kantarcioglu, M., & Vaidya, J. (2002). *Defining Privacy for Data Mining.* Paper presented at the the National Science Foundation Workshop on Next Generation Data Mining, Baltimore, MD.

Du, W., & Atallah, M. J. (2001). *Privacy-preserving cooperative statistical analysis.* Paper presented at the Proceedings 17th Annual Computer Security Applications Conference, New Orleans, LA, USA.

Du, W., Han, Y. S., & Chen, S. (2004). *Privacy-preserving multivariate statistical analysis: Linear regression and classification.* Paper presented at the Proceedings of the 4th SIAM International Conference on Data Mining, Lake Buena Vista, FL, United States.

Du, W., & Zhan, Z. (2002). *Building decision tree classifier on private data.* Paper presented at the Proceedings of the IEEE International Conference on Privacy, Security and Data Mining, Maebashi City, Japan.

Goldreich, O. (2002). Secure multi-party computation: manuscript.

Goldwasser, S., Micali, S., & Rackoff, C. (1989). The knowledge complexity of interactive proof systems. *SIAM Journal on Computing, 18*(1), 186-208.

Han, S., & Ng, W. K. (2007a). *Privacy-Preserving Genetic Algorithms for Rule Discovery.* Paper presented at the the 9th International Conference on Data Warehousing and Knowledge Discovery (DaWaK 2007) Regensburg, Germany

Han, S., & Ng, W. K. (2007b). *Privacy-Preserving Self-Organizing Map.* Paper presented at the the 9th International Conference on Data Warehousing and Knowledge Discovery (DaWaK 2007) Regensburg, Germany

Jagannathan, G., Pillaipakkamnatt, K., & Wright, R. N. (2006). *A new privacy-preserving distributed k-clustering algorithm.* Paper presented at the Proceedings of the SIAM International Conference on Data Mining, Bethesda, MD, United States.

Jagannathan, G., & Wright, R. N. (2005). *Privacy-preserving distributed k-means clustering over arbitrarily partitioned data.* Paper presented at the Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, IL, United States.

Kantarcioglu, M., & Clifton, C. (2003). *Privacy preserving naive bayes classifier for horizontally partitioned data.* Paper presented at the IEEE ICDM Workshop on Privacy Preserving Data Mining, Melbourne, FL.

Kantarcioglu, M., & Clifton, C. (2004). Preserving data mining of association rules on horizontally partitioned data. *Transactions on Knowledge and Data Engineering, 16*(9), 1026-1037.

Laur, S., Lipmaa, H., & Mielikainen, T. (2006). *Cryptographically private support vector machines.* Paper presented at the Proceedings of the 12th ACM International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA.

Lin, X., Clifton, C., & Zhu, M. (2005). Privacy-preserving clustering with distributed EM mixture modeling. *Knowledge and Information Systems, 8*(1), 68 - 81.

Lindell, Y., & Pinkas, B. (2000). *Privacy preserving data mining.* Paper presented at the Proceedings of Advances in Cryptology, Santa Barbara, CA, USA.

Lindell.Y., & Pinkas, B. (2000). *Privacy preserving data mining.* Paper presented at the Advances in Cryptology.

Vaidya, J., & Clifton, C. (2002). *Privacy preserving association rule mining in vertically partitioned data.* Paper presented at the Proceedings of the 8th ACM International Conference on Knowledge Discovery and Data Mining, Edmonton, Alberta, Canada.

Vaidya, J., & Clifton, C. (2003). *Privacy-preserving k-means clustering over vertically partitioned data.* Paper presented at the Proceedings of the 9th ACM International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA.

Vaidya, J., & Clifton, C. (2004). *Privacy preserving naive bayes classifier for vertically partitioned data.* Paper presented at the Proceedings of the SIAM International Conference on Data Mining, Lake Buena Vista, Florida.

Vaidya, J., & Clifton, C. (2005a). *Privacy-preserving decision trees over vertically partitioned data.* Paper presented at the Proceedings of the 19th Annual IFIP WG 11.3 Working Conference on Data and Applications Security, Storrs, Connecticut.

Vaidya, J., & Clifton, C. (2005b). Secure set intersection cardinality with application to association rule mining. *Journal of Computer Security, 13*(4).

Vaidya, J., Kantarcioglu, M., & Clifton, C. (2007). Privacy-preserving Naive Bayes Classification. *The VLDB Journal*.

Wan, L., Ng, W. K., Han, S., & Lee, V. (2007). *Privacy-preservation for gradient descent methods.* Paper presented at the Proceedings of the Thirteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, California.

Wright, R., & Yang, Z. (2004). *Privacy-preserving bayesian network structure computation on distributed heterogeneous data.* Paper presented at the Proceedings of the 10th ACM International Conference on Knowledge Discovery and Data Mining, Seattle, WA, USA.

Yao, A. C. (1986). *How to generate and exchange secrets.* Paper presented at the Proceedings of the Annual IEEE Symposium on Foundations of Computer Science.

Yu, H., Jiang, X., & Vaidya, J. (2006). *Privacy-preserving svm using nonlinear kernels on horizontally partitioned data.* Paper presented at the Proceedings of the ACM Symposium on Applied Computing, Dijon, France.

Yu, H., Vaidya, J., & Jiang, X. (2006). *Privacy-preserving svm classification on vertically partitioned data.* Paper presented at the Proceedings of the 10th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Singapore.

KEY TERMS

Horizontally Partitioned Data: In horizontally partitioned data, two or more parties hold different disjoint set of objects (tuples) all having the same set of attributes.

Malicious Model: A party in the malicious model is allowed to behave arbitrarily and deviate from the protocol.

Privacy-Preserving Data Mining: Privacy-preserving data mining seeks to allow the cooperative execution of data mining algorithms while preserving the data privacy of each party concerned.

Randomization: A party randomizes its private value by adding to it a random shift. The shift values are independently and identically distributed random variables.

Secure Multi-Party Computation: A set of parties each holding private inputs wishes to jointly compute a function or a computation of their inputs. After the computation, they only know the correct joint output and nothing about the other parties' private data.

Semi-Honest Model: Stated informally, a party in the semi-honest model follows and executes a designated protocol properly. However, it may keep a record of the intermediate results of the execution and use them to compute additional information about the private inputs of other honest parties.

Vertically Partitioned Data: In vertically partitioned data, two or more parties hold different disjoint set of attributes for the same set of objects (tuples).

Secure Computation for Privacy Preserving Data Mining

Yehuda Lindell
Bar-Ilan University, Israel

INTRODUCTION

The increasing use of data mining tools in both the public and private sectors raises concerns regarding the potentially sensitive nature of much of the data being mined. The utility to be gained from widespread data mining seems to come into direct conflict with an individual's need and right to privacy. Privacy preserving data mining solutions achieve the somewhat paradoxical property of enabling a data mining algorithm to use data *without* ever actually "seeing" it. Thus, the benefits of data mining can be enjoyed, without compromising the privacy of concerned individuals.

BACKGROUND

A classical example of a privacy preserving data mining problem is from the field of medical research. Consider the case that a number of different hospitals wish to jointly mine their patient data, for the purpose of medical research. Furthermore, let us assume that privacy policy and law prevents these hospitals from ever pooling their data or revealing it to each other due to the confidentiality of patient records. In such a case, classical data mining solutions cannot be used. Fortunately, privacy preserving data mining solutions enable the hospitals to compute the desired data mining algorithm on the union of their databases, without ever pooling or revealing their data. Indeed, the only information (provably) learned by the different hospitals is the output of the data mining algorithm. This problem whereby different organizations cannot directly share or pool their databases, but must nevertheless carry out joint research via data mining, is quite common. For example, consider the interaction between different intelligence agencies in the USA. These agencies are suspicious of each other and do not freely share their data. Nevertheless, due to recent security needs, these agencies must run data mining algorithms on their combined data. Another example relates to data that is held by governments. Until recently, the Canadian Government held a vast federal database that pooled citizen data from a number of different government ministries (this database was called the "big brother" database by some). The Canadian government claimed that the database was essential for research. However, due to privacy concerns and public outcry, the database was dismantled, thereby preventing that "essential research" from being carried out. This is another example of where privacy preserving data mining could be used to balance between real privacy concerns and the need of governments to carry out important research.

Privacy preserving data mining is actually a special case of a long-studied problem in cryptography: *secure multiparty computation*. This problem deals with a setting where a set of parties with private inputs wish to jointly compute some function of their inputs. Loosely speaking, this joint computation should have the property that the parties learn the correct output and nothing else, even if some of the parties maliciously collude to obtain more information.

MAIN THRUST

In this short chapter, we will provide a succinct overview of secure multiparty computation, and how it can be applied to the problem of privacy preserving data mining. Our main focus will be on how security is formally defined, why this definitional approach is adopted, and what issues should be considered when defining security for privacy preserving data mining problems. Due to space constraints, the treatment in this chapter is both brief and informal. For more details, we refer the reader to (Goldreich, 2003) for a survey on cryptography and cryptographic protocols.

Security Definitions for Secure Computation

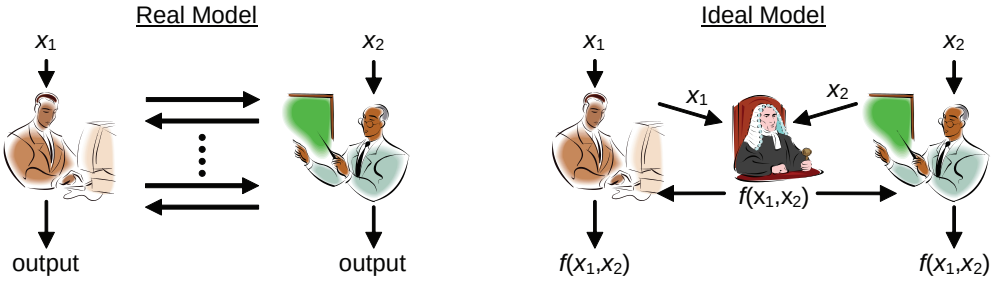
The aim of a secure multiparty computation task is for the participating parties to *securely* compute some function of their distributed and private inputs. However, what does it mean for a computation to be *secure*? One way of approaching this question is to provide a list of *security properties* that should be preserved. The first such property that often comes to mind is that of *privacy* or *confidentiality*. A naïve attempt at formalizing privacy would be to require that each party learns nothing about the other parties' inputs, even if it behaves maliciously. However, such a definition is usually unattainable because the defined output of the computation itself typically reveals some information on other parties' inputs. (For example, a decision tree computed on two distributed databases reveals some information about both databases.) Therefore, the privacy requirement is usually formalized by saying that the only information learned by the parties in the computation (again, even by those who behave maliciously) is that specified by the function output. Although privacy is a primary security property, it rarely suffices. Another important property is that of *correctness*; this states the honest parties' outputs are correctly distributed even in the face of adversarial attack. A central question that arises in this process of defining security properties is: *when is our list of properties complete*? This question is, of course, application-dependent and this essentially means that for every new problem, the process of deciding which security properties are required must be re-evaluated. We stress that coming up with the right list of properties is often very difficult, and it can take many years until we are convinced that a definition truly captures the security requirements that are needed. Furthermore, an incomplete of properties may easily lead to real security failures.

The Ideal/Real Model Paradigm

Due to these difficulties, the standard definitions of secure computation today follow an alternative approach called the *ideal/real model paradigm*. This has been the dominant paradigm in the investigation of secure computation in the last fifteen years; we refer the reader to (Canetti, 2000) for the formal definition and references therein for related definitional work. Loosely speaking, this paradigm defines the security of a real protocol by comparing it to an *ideal computing scenario* in which the parties interact with an external trusted and incorruptible party. In this ideal execution, the parties all send their inputs to the trusted party (via ideally secure communication lines). The trusted party then computes the function on these inputs and sends each party its specified output. Such a computation embodies the goal of secure computation, and it is easy to see that the properties of privacy and correctness hold in the ideal model. In addition to the fact that these and other security properties are preserved in an ideal execution, the simplicity of the ideal model provides an intuitively convincing security guarantee. For example, notice that the only message that a party sends in an ideal execution is its input, and so the only power that a corrupted party has is to choose its input (something which is typically legitimate anyway).

So far, we have defined an ideal execution in an ideal world. However, in the *real world*, the parties run a protocol without any trusted help. Despite this, a secure real protocol should somehow "emulate" an ideal execution. That is, we say that a real protocol that is run by the parties (in a world where no trusted party exists) is ***secure***, if *no adversary* can do more harm in

Box 1.



a real execution than in an execution that takes place in the ideal world. Stated differently, for *any adversary* carrying out a successful attack on a real protocol, there exists an adversary that successfully carries out the same attack in the ideal world. This suffices because, as we have seen, *no successful attacks* can be carried out in an ideal execution. Thus, no successful attacks can be carried out on the real protocol, implying that it is secure. See Box 1 for a diagram of the real and ideal models.

We stress that security is required to hold for *every* adversary carrying out *any feasible attack* (within the parameters defined for the adversary, as discussed next).

Defining the Model

The above informal description of the ideal/real model paradigm expresses the intuition that a real execution should behave just like an ideal execution. In order to obtain a complete and formal definition, it is crucial that both the ideal and real models are fully defined. Among other things, this involves defining the real network model and the adversary's power, including any assumptions on its behavior. We stress that a secure protocol only provides real-world security guarantees if the mathematical definition of the real computation and adversarial models accurately reflects the real network and adversarial threats that exist.

We now briefly discuss a number of parameters that are considered when defining the network model and the adversary (this list is far from comprehensive). Two central considerations that arise when defining the network model relate to the communication channels and whether or not any trusted setup phase is assumed. It is typically assumed that all parties are connected via point-to-point *authenticated* channels (meaning that the adversary cannot modify messages sent between honest parties). We note that this can be implemented assuming a public-key infrastructure for digital signatures. Other parameters to consider are whether the communication over the network is synchronous or asynchronous, and whether or not messages that are sent between honest parties are guaranteed to arrive. Finally, the question of what (if any) other protocols are running simultaneously in the network must also be addressed. This issue is referred to as *protocol composition* and is currently a very active research subject in the cryptographic community. When defining the adversary, a number of possibilities arise. We now describe some of these:

1. **Complexity:** Given that the widely accepted notion of efficient or feasible computation is *probabilistic polynomial-time*, the natural choice is to limit an adversary to this complexity. However, there are also protocols that are secure against *unbounded* adversaries.
2. **Number of corrupted parties:** In a general multiparty setting, we assume that the adversary controls some subset of the participating parties; these parties are called corrupted. The allowed size of this subset must also be defined (typical choices are assuming that less than one third or one half are corrupted, and not assuming any limitation on the number of corrupted parties).
3. **Corruption strategy:** This parameter relates to whether or not the adversary is *static* (meaning that the set of corrupted parties is fixed ahead of time), or *adaptive* (meaning that the adversary can "break into" parties during the protocol execution).
4. **Allowed adversarial behavior:** In our discussion above, we implicitly referred to *malicious* adversaries who are allowed to arbitrarily deviate from the protocol specification. However, the adversary's behavior is sometimes restricted. For example, a *semi-honest* adversary is assumed to follow the protocol but may attempt to learn secret information from the messages that it receives.

The above very partial list of parameters for defining the adversary begs the question: how do we decide which adversarial model to take? A conservative approach is to take the most powerful adversary possible. However, being overly conservative comes at a price. For example, it is impossible to obtain security for unbounded adversaries in the case that half or more of the parties are corrupted. Furthermore, it is often the case that more efficient protocols can be constructed for weaker adversaries (specifically, highly efficient protocols for many tasks are known for the semi-honest adversarial model, but this is not the case for the malicious model). In general, a good approach is to consider malicious polynomial-time adversaries who may adaptively corrupt any number of the participants. However, in some cases, the semi-honest adversarial model is reasonable. For example, in the medical data-

base example provided in the Introduction, the hospitals are not believed to be malicious; rather, the law prevents them from revealing confidential patient data. In such a case, the protection provided by semi-honest adversarial modeling is sufficient. We stress, however, that in many cases the semi-honest model is not realistic, and malicious adversaries must be considered.

In summary, two central guiding principles when defining security are that: **(a)** the definition must accurately and conservatively model the real-world network setting and adversarial threats, and **(b)** all aspects of the model must be fully and explicitly defined. These conditions are necessary for obtaining a mathematical definition of security that truly implies that protocols executed in the real world will withstand all adversarial attacks.

The Feasibility of Secure Multiparty Computation

The aforementioned security definition provides very strong guarantees. An adversary attacking a protocol that is secure is essentially limited to choosing its input (because this is all that it can do in the ideal model). However, can this definition actually be achieved, and if yes under what conditions? A fundamental result of the theory of cryptography states that under certain parameters and assumptions, *any* efficient multiparty functionality can be securely computed. This result is comprised of a number of different theorems, depending on the model and the number of corrupted parties. We will describe the basic results for the stand-alone model (where only a single protocol execution is considered), and the computational setting (where the adversary is limited to polynomial-time). The basic results for the information-theoretic setting can be found in (Ben-Or et al., 1988) and (Chaum et al., 1988).

The first basic theorem states that when a majority of the parties are honest, *any multiparty functionality* can be securely computed in the presence of malicious, static adversaries (Yao, 1986; Goldreich et al., 1986). Extensions to the case of adaptive adversaries can be found in (Beaver and Haber, 1992) and (Canetti et al., 1996).

The second basic theorem relates to the case that any number of parties may be corrupted, and so there is not necessarily an honest majority. In this case, it is impossible to construct protocols that meet the definition as described above. The reason for this is that the definition implies that all parties receive output together; however, this cannot be achieved without an honest majority (Cleve, 1986). We therefore explicitly relax the security definition and allow the adversary to prevent the honest parties from receiving their output, even in the ideal model; this relaxed definition is called *security with abort*. As above, it has been shown that even when any number of parties may be corrupted, *any multiparty functionality* can be securely computed in the presence of malicious, static adversaries (Yao, 1986; Goldreich et al., 1986).

As we have mentioned, the above results all refer to the *stand-alone model* of computation, where it is assumed that the secure protocol being analyzed is run once in isolation. Feasibility results have also been shown for the case of protocol composition where many different protocols run concurrently; for example, see (Canetti, 2001) and (Canetti et al., 2002). A brief survey on known results for the setting of composition can be found in (Lindell, 2003).

The importance of the above results is that they demonstrate that under an appropriate choice of parameters and assumptions, *any* privacy preserving data mining problem can be solved, *in principle*. Therefore, the challenge remaining is to construct protocols that are efficient enough for practical use.

Secure Protocols for Privacy Preserving Data Mining

The first paper to take the classic cryptographic approach to privacy preserving data mining was (Lindell and Pinkas, 2002). The paper presents an efficient protocol for the problem of distributed decision tree learning; specifically, how to securely compute an ID3 decision tree from two private databases. The paper considered *semi-honest* adversaries only. This approach was adopted in a relatively large number of works that demonstrate semi-honest protocols for a wide variety of data mining algorithms; see, for example, (Clifton et al., 2003). In our opinion, these results serve as a "proof of concept" that highly efficient protocols can be constructed, even for seemingly complex functions. However, in many cases, the semi-honest adversarial model does *not* suffice. Therefore, the malicious model must also be considered.

Other work on the problem of privacy preserving data mining has followed what is often called the "data perturbation" approach, as introduced by (Agrawal &

Srikant, 2000). We remark that the development of rigorous security definitions that appropriately model security in settings considered by this approach seems to be a very difficult task. We remark that naïve definitions of security have been shown to be completely insecure; see (Dinur and Nissim, 2003) for just one example.

FUTURE TRENDS

As we have shown, there exist secure solutions for all privacy preserving data mining problems. However, these solutions are usually not efficient enough for use in practice. Thus, the main problem of privacy preserving data mining is to find protocols that can realistically be run, even on very large databases. Until now, most work has focused on the semi-honest adversarial model, which often does not provide a sufficient level of security. It is therefore of great importance to begin developing tools and techniques for constructing highly efficient protocols that are secure against malicious adversaries. We note that achieving this goal may involve finding new definitions that are more relaxed than those described here, yet still accurately model the real security concerns. This is a very non-trivial research task due to the subtle nature of security and security definitions.

CONCLUSIONS

The history of cryptography shows very clearly that when protocols are not proven secure, or when the adversarial models are not explicitly defined, real attacks are very often discovered. Furthermore, the task of coming up with mathematical definitions that accurately model real adversarial threats is a very difficult task. Indeed, slight modifications to existing definitions can render them completely useless; see (Canetti, 2000) for some discussions on this issue. In this short article, we have described the real/ideal model paradigm for defining security. This definitional approach has been the fruit of many years of cryptographic research, and protocols that meet this definition provide very powerful security guarantees. In order to provide efficient solutions for privacy preserving data mining problems, it may be necessary to find new definitions that provide both rigorous security guarantees and can be met by highly efficient protocols. This is perhaps the ultimate challenge of this new and exciting field of research.

REFERENCES

Agrawal, R., and Srikant, R. (2000). Privacy Preserving Data Mining. *ACM SIGMOD International Conference on Management of Data*, SIGMOD'00, Dallas, USA. 439-450.

Beaver, D. & Haber, S. (1992). Cryptographic Protocols Provably Secure Against Dynamic Adversaries. *Advances in Cryptology*, EUROCRYPT'92. Balatonfüred, Hungary, 307-323.

Ben-Or, M., Goldwasser, S. & Wigderson, A. (1988). Completeness Theorems for Non-Cryptographic Fault-Tolerant Distributed Computation. The *20th Annual ACM Symposium on Theory of Computing*, STOC'88. Illinois, USA, 1-10.

Canetti, R. (2000). Security and Composition of Multiparty Cryptographic Protocols. *Journal of Cryptology*, 13 (1), 143-202.

Canetti, R. (2001). Universally Composable Security: A New Paradigm for Cryptographic Protocols. The *42rd Annual IEEE Symposium on the Foundations of Computer Science*, FOCS'01. Nevada, USA, 136-145.

Canetti, R., Lindell, Y., Ostrovsky, R. & Sahai, A. (2002). Universally Composable Two-Party and Multi-Party Computation. The *34th Annual ACM Symposium on Theory of Computing*, STOC'02. Montreal, Canada, 494-503.

Canetti, R., Feige, U., Goldreich, O. & Naor, M. (1996). Adaptively Secure Multi-Party Computation. The *28th Annual ACM Symposium on Theory of Computing*, STOC'96. Pennsylvania, USA, 639-648.

Chaum, D., Crepeau, C. & Damgard, I. (1988). Multiparty Unconditionally Secure Protocols. The *20th Annual ACM Symposium on Theory of Computing*, STOC'88. Illinois, USA, 11-19.

Cleve, R. (1986). Limits on the Security of Coin Flips when Half the Processors are Faulty. The *18th Annual ACM Symposium on Theory of Computing*, STOC'86. California, USA, 364-369.

Clifton, C., Kantarcioglu, M., Vaidya, J., Lin, X. & Zhu, M.Y. (2003). Tools for Privacy Preserving Data Mining. *SIGKDD Explorations*, 4 (2), 28-34.

Dinur, I. and Nissim, K. (2003). Revealing Information While Preserving Privacy. *ACM Symposium on Principles of Database Systems*, PODS'03. San-Diego, USA, 202-210.

Goldreich, O. (2003). Cryptography and Cryptographic Protocols. *Distributed Computing*, 16 (2), 177-199.

Goldreich, O., Micali, S. & Wigderson A. (1987). How to Play any Mental Game – A Completeness Theorem for Protocols with Honest Majority. The *19th Annual ACM Symposium on the Theory of Computing*, STOC'87. New York, USA, 218-229.

Lindell, Y. (2003). *Composition of Secure Multi-Party Protocols – A Comprehensive Study.* Springer-Verlag.

Lindell, Y. & Pinkas, B. (2002). Privacy Preserving Data Mining. *Journal of Cryptology*, 15 (3), 177-206.

Yao, A. (1986). How to Generate and Exchange Secrets. The *27th Annual IEEE Symposium on the Foundations of Computer Science*, FOCS'86. Toronto, Canada, 162-167.

KEY TERMS

Corrupted Parties: parties that participate in a protocol while under the control of the adversary.

Functionality: the task that the parties wish to jointly compute.

Ideal Model: a virtual setting where all parties interact with an incorruptible trusted party who carries out the joint computation for them.

Malicious Adversary: an adversary who may arbitrarily deviate from the protocol specification (and so is unlimited in its attack strategy).

Real Model: the setting where a real protocol is run (without any trusted help).

Secure Multiparty Computation: the problem of computing any distributed task so that security is preserved in the face of adversarial attacks.

Semi-Honest Adversary: an adversary who follows the protocol specification, but may try to learn private information by analyzing the messages that it receives during the protocol execution. (This models the inadvertent leakage of information even during legitimate protocol executions.)

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 1005-1009, copyright 2005 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

Segmentation of Time Series Data

Parvathi Chundi
University of Nebraska at Omaha, USA

Daniel J. Rosenkrantz
University of Albany, SUNY, USA

INTRODUCTION

Time series data is usually generated by measuring and monitoring applications, and accounts for a large fraction of the data available for analysis purposes. A time series is typically a sequence of values that represent the state of a variable over time. Each value of the variable might be a simple value, or might have a composite structure, such as a vector of values. Time series data can be collected about natural phenomena, such as the amount of rainfall in a geographical region, or about a human activity, such as the number of shares of Google™ stock sold each day. Time series data is typically used for predicting future behavior from historical performance. However, a time series often needs further processing to discover the structure and properties of the recorded variable, thereby facilitating the understanding of past behavior and prediction of future behavior. Segmentation of a given time series is often used to compactly represent the time series (Gionis & Mannila, 2005), to reduce noise, and to serve as a high-level representation of the data (Das, Lin, Mannila, Renganathan & Smyth, 1998; Keogh & Kasetty, 2003). Data mining of a segmentation of a time series, rather than the original time series itself, has been used to facilitate discovering structure in the data, and finding various kinds of information, such as abrupt changes in the model underlying the time series (Duncan & Bryant, 1996; Keogh & Kasetty, 2003), event detection (Guralnik & Srivastava, 1999), etc.

The rest of this chapter is organized as follows. The section on **Background** gives an overview of the time series segmentation problem and solutions. This section is followed by a **Main Focus** section where details of the tasks involved in segmenting a given time series and a few sample applications are discussed. Then, the **Future Trends** section presents some of the current research trends in time series segmentation and the **Conclusion** section concludes the chapter. Several important terms and their definitions are also included at the end of the chapter.

BACKGROUND

A time series is simply a sequence of data points. In a segmentation of a given time series, one or more consecutive data points are combined into a single *segment* and represented by a single data point or a model for the data points in the segment. Given a time series T of n data points, segmentation of T results in a sequence of m segments, where each segment represents one or more consecutive data points in T. The number of segments m is typically much less than n. The input to a given instance of the segmentation problem is usually a time series and an upper bound on the number of segments.

Since the data points in a segment are represented by a single point or model, there is usually some error in the representation. Formally, the segmentation problem can be defined as follows. Given time series T and integer m, find a minimum error segmentation of T consisting of at most m segments. A specific version of this problem depends on the form of the data and how the segmentation error is defined. There are several approaches in the literature for addressing the segmentation problem. An *optimum* solution for the segmentation problem can be found by a dynamic programming based approach. (Duncan & Bryant, 1996; Gionis & Mannila, 2005; Himberg, Korpiaho, Mannila, Tikanmaki & Toivonen, 2001). A dynamic programming algorithm for solving this optimization problem runs in $O(n^2m)$ time, and so may not be practical for long time series with thousands of data points. There are more efficient heuristics that can be used to construct a segmentation of a given time series. However, these heuristics generally produce a suboptimal segmentation (Himberg, Korpiaho, Mannila, Tikanmaki & Toivonen,

2001; Keogh, Chu, Hart & Pazzani, 2004). There are also Bayesian based, fuzzy clustering, and genetic algorithm approaches to the segmentation problem (Oliver & Forbes, 1997; Abonyi, Feil, Nemeth, & Arva, 2005; Tseng, Chen, Chen & Hong, 2006). Methods have also been developed where a time series is segmented by converting it into a sequence of discrete symbols (Chung, Fu, Ng & Luk, 2004).

References (Chundi & Rosenkrantz, 2004a; Chundi & Rosenkrantz, 2004b) discuss segmentation algorithms for time series where each data point is a set of documents, each containing a set of keywords or key phrases. Reference (Siy, Chundi, Rosenkrantz & Subramaniam, 2007) gives an application of segmentation for time series where each data point is a set of items. References (Cohen, Heeringa & Adams, 2002; Gionis & Mannila, 2005) discuss segmentation algorithms for time series where each data point is a single symbol from an alphabet.

MAIN FOCUS

The main focus of a segmentation algorithm is to find the best segmentation to represent the given time series. There are *two primary* tasks in the segmentation process: constructing the representation of a given segment, and constructing the overall segmentation.

Representation of a Given Segment

The data representation of a given segment is computed from the time series data points in the segment, and can be viewed as representing a model of a process that may have generated these data points. The representation of a segment depends on the type of the data points in the time series, and the kind of model to be used. The usual goal in constructing the representation of a given segment is to minimize the error between the representation and the data points in the segment. Each data point in a time series can be a single numeric value (Keogh & Kasetty, 2003; Keogh, Chu, Hart & Pazzani, 2004), a vector of numeric values (McCue & Hunter, 2004), a set of items (Siy, Chundi, Rosenkrantz & Subramaniam, 2007) , a set of documents (Chundi & Rosenkrantz, 2004a), or some other type of value.

When each time series data point is a single numeric value, the model underlying a segment is typically a curve fitted to the points in the segment. This curve is constructed from the data points in the segment, usually with the goal of minimizing the error between the sequence of values of the data points in the segment, and the sequence of values implied by the segment representation. This segment error is usually computed by combining the difference between the value of each data point and the value assigned to that time point by the segment representation. These differences are often combined using an L_p metrics. L_1 is the Manhattan distance, i.e., the sum of the magnitude of the differences. L_2 is the Euclidean distance, based on the sum of the squares of the differences. In addition to the L_p metrics, local PCA methods have been used to measure error (Abonyi, Feil, Nemeth & Arva, 2005).

The curve for a given segment may be a single number (piecewise constant representation), a linear function (piecewise linear representation), a quadratic function, or a polynomial of even higher degree. A piecewise constant representation is a single numeric value that minimizes the segment error. For the L_1 error metric, this value is the median of the data points in the segment, and for the L_2 metric, it is the average of the data points (Gionis & Mannila, 2005). A piecewise linear representation fits a straight line to the data points (Keogh & Kasetty, 2003), usually with the goal of minimizing the L_2 error metric (Edwards, 1976). A piecewise constant or piecewise linear representation can be computed more efficiently than a quadratic or higher degree representation (Lamire, 2007). Sometimes the segments in a segmentation have different types of representation; e.g., some segments may have a linear representation, and some may have a quadratic representation, etc. (Lamire, 2007; Pednault, 1991).

When each data point is a vector of numeric values, a segment may be represented as a vector of constant values (McCue & Hunter, 2004), a vector of line segments (Abonyi, Feil, Nemeth & Arva, 2005), a vector of quadratics, etc.

When each data point is a set of documents, each of which contains a set of keywords or key phrases, the segment error is a measure of how closely the keywords (or key phrases) for the segment correspond to those for the documents in the time points of the segment (Chundi & Rosenkrantz, 2004a; Chundi & Rosenkrantz, 2004b).

Best Plan for Overall Segmentation

Several parameters must be specified to construct a segmentation for a given time series. Some of the commonly provided parameters are the desired number of segments, any constraints on individual segments, and the optimality criterion. An aspect ratio constraint may also be provided, specifying an upper bound on ratio of the length of the segmentation's longest segment to the length of its shortest segment.

A segmentation of a time series consists of multiple segments, and each segment error contributes to the overall error in segmentation. This overall segmentation error may be defined to be the sum of the individual segment errors, an average of these errors, or the maximum of these errors. It is also possible that future applications will consider some other means of combining the segment errors.

The simplest segmentation that can be built for a given time series is a *uniform segmentation*, where all segments are of the same size (Lavrenko, Schmill, Lawrie, Ogilvie, Jensen & Allan, 2000; Lent, Agrawal & Srikant, 1997). A straightforward way to construct a uniform segmentation is to combine all data points that fall in a given period of time (such as the same day, week, month, quarter, year, etc.) into a single segment. Or, given a specification that there should be m segments, one can make each of these segments be a sequence of n/m consecutive points from the time series (with suitable modification if n/m is not an integer).

The segmentation optimization problem requires construction of a segmentation that minimizes the overall segmentation error. Since a uniform segmentation is constructed without looking at the actual data values in the time series, there is no particular reason why it would minimize the overall error, or even be a good approximation to an optimum segmentation. Indeed, it is easy to construct time series where a uniform segmentation has an arbitrarily larger overall segmentation error than an optimum segmentation with the same number of segments.

Several methods have been developed to construct an optimum segmentation. In addition, several heuristics have been developed that do not necessarily construct an optimum segmentation, but construct acceptably good segmentations in many cases. Further research is needed to provide quality of approximation guarantees or otherwise characterize how well the segmentations produced by these heuristics do in terms of overall segmentation error, in comparison with an optimum segmentation.

Dynamic programming based schemes are commonly employed to construct optimal segmentations (Duncan & Bryant, 1996; Himberg, Korpiaho, Mannila, Tikanmaki & Toivonen, 2001; Keogh & Kasetty, 2003; Pavlidis & Horowitz, 1974). Given a time series T with n data points, an upper bound m on the number of segments, and the segment error of each possible segment of T, a dynamic programming based scheme constructs a segmentation whose overall error is minimal. Let (i, j) denote the segment of T consisting of data points i through j. The scheme typically maintains a table R to record the minimal error value possible in covering time points 1 through j with k segments. (Variable k ranges between 1 and the maximum of m and j.) For $k = 1$, entry $R[j, 1]$ of the table is set to the segment error of segment $(1, j)$. For $k > 1$, entry $R[j, k]$ is constructed from other values in R. In the following example, the optimization goal is to minimize the sum of the individual segment errors. Accordingly, $R[j, k]$ is set based on the idea that if in an optimum k segment covering of time points 1 through j, the first $k-1$ segments cover time points 1 through x, then these segments provide a minimum total cost covering of these time points by $k-1$ segments.

$$R[j, k] = min_{k-1 \le x < j} \{R[x, k-1] + \text{segment error of segment } (x+1, j)\}$$

The Reference (Chundi & Rosenkrantz, 2004b) presents a dynamic programming algorithm where the parameters include a constraint on the size, and/or variability of the size, of segments in a segmentation.

Dynamic programming based segmentation methods run in time $O(n^2m)$ time, where n is the number of data points in the time series and m is the desired number of segments. For many applications, where time series data may contain hundreds of thousands of data points, dynamic programming based methods are too slow. In such cases, a user may opt for suboptimal segmentations. There are heuristic methods that take less time than dynamic programming methods to produce a segmentation. Three popular heuristic schemes are sliding window, top-down, and bottom-up methods (Keogh, Chu, Hart & Pazzani, 2004; Lamire, 2007). These heuristic methods save time by considering only

some possible segments to hopefully find a good segmentation, whereas dynamic programming considers all $O(n^2)$ segments to find an optimal segmentation. Consequently, heuristic based methods may not find the best segmentation.

In addition to the above methods, fuzzy clustering based methods have also been studied to construct segmentations of time series data (Abonyi, Feil, Nemeth & Arva, 2005) where segments are identified as fuzzy sets and the hyper planes of local PCA models are used to measure segment errors.

Sample Applications

Segmentation methods have been employed in many domains. We describe several recent applications. There has been recent interest in temporal analysis of software version histories. Such histories contain a wealth of information about development activities of a software project. A software version history contains, among other information, the identity of files that have changed at a given time point and the developers that made the changes (Siy, Chundi, Rosenkrantz & Subramaniam, 2007). This information is represented as a time series where each data point is a set of discrete items, such as files and developers. Then, several dynamic-programming based segmentations as well as uniform segmentations of varying sizes the time series are constructed. Dynamic-programming based segmentation performed better by capturing more of the temporal content of the time series than uniform segmentations. Time segments in a dynamic-programming based segmentation corresponded to natural stages in the history of the paper. In addition, the study identified developers that were actively changing files in a time segment and discovered that changes to files made by active developers in a time segment leads to fewer fixes in the future.

Segmentation methods have also been employed to identify temporal information from unstructured text documents (Chundi & Rosenkrantz, 2004a; Chundi & Rosenkrantz, 2004b). The time of creation/publication is used to create a time series over documents, where each data point is a set of documents and keywords (or key phrases) occurring in those documents. Segmentation of a document set time series has been used to remove noise from data and analyze trends in topics.

FUTURE TRENDS

Segmentation of time series data continues to be an active area of research, as it is central to understanding and mining temporal data. We outline a few recent research issues here. Time series segmentation methods are being extended to applications where data points are non-numeric; they may be sets of discrete items, sets of sets, or some kind of a nested structure. Definitions of segment errors, overall errors and other functions appropriate to various applications must be suitably defined, so that current segmentation methods can be adapted to new kind of data. Traditionally a time series is considered to consist of a finite number of observations, all available before a segmentation is constructed. However, it is often more natural to think of a time series as an unending stream of observations, obtained as a natural or man-made phenomenon is monitored. For such streaming data, a segmentation may need to be constructed incrementally, as observations become available. Finally, methods are being designed where the polynomial degree of different segment representations in a segmentation may vary, with a higher degree polynomial used only for those segments where it has a major effect on reducing the overall error.

CONCLUSION

Time series data is pervasive in many domains, including financial, scientific, and healthcare domains, and is very useful for understanding past information and predicting future trends. Time series segmentation is an important problem since data mining methods are often applied to a segmented time series to discover useful information. Therefore, it is important to ensure that the segmentation of a time series provides an accurate representation of the underlying time series data. Time series segmentation is still an active area of research, including developing approaches for streaming data and scaling segmentation algorithms to long time series (with millions of data points).

REFERENCES

Abonyi, J., Feil, B., Nemeth, S., & Arva, P. (2005), Modified Gath-Geva Clustering for Fuzzy Segmen-

tation of Multivariate Time-series, *Fuzzy Sets and Systems*, 149(1), 39-56.

Chundi, P., & Rosenkrantz, D. J. (2004a), Constructing Time Decompositions for Time Stamped Documents, *SIAM Fourth International Conference on Data Mining*, 57-68.

Chundi, P., & Rosenkrantz, D. J. (2004b), On Lossy Time Decompositions of Time Stamped Documents, *2004 ACM CIKM Conference on Information and Knowledge Management*, 437-445.

Chung, F., Fu, T., Ng, V., & Luk, R. W. P. (2004), An Evolutionary Pattern Based Method for Time Series Segmentation, *IEEE Transactions on Evolutionary Computation*, 8(5), 471-489.

Cohen, P. R., Heeringa, B., & Adams, M. N. (2002), Unsupervised Segmentation of Categorical Time Series into Episodes, *Proceedings of the 2002 IEEE International Conference on Data Mining*, 99-106.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2002), *Introduction to Algorithms*, McGraw-Hill Publishers.

Das, G., Lin, K., Mannila, H., Renganathan, G., & Smyth, P. (1998), Rule Discovery from Time Series, *The Fourth International Conference on Knowledge Discovery and Data Mining*, 16-22.

Duncan, S. R. & Bryant, G. F. (1996) , A New Algorithm for Segmenting Data from Time Series, *The 35th Conference on Decision and Control*, 3123-3128.

Edwards, A. L., (1976), *An Introduction to Linear Regression and Correlation*, W. H. Freeman & Co Ltd.

Gionis, A., & Mannila, H. (2005), Segmentation Algorithms for Time Series and Sequence Data, A Tutorial in the *SIAM International Conference on Data Mining*.

Guralnik, V., & Srivastava, J. (1999), Event Detection from Time Series Data, *Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 33-42.

Himberg, J., Korpiaho, K., Mannila, H., Tikanmaki, J., & Toivonen, H. (2001), Time Series Segmentation for Context Recognition in Mobile Devices, *IEEE International Conference on Data Mining*, 203-210.

Keogh, E. J., & Kasetty, S. (2003), On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration, *Journal of Data Mining and Knowledge Discovery*, 7(4), 349-371.

Keogh, E. J., Chu, S., Hart, D., & Pazzani, M. (2004), Segmenting Time Series: A Survey and Novel Approach, *Data Mining in Time Series Databases*, World Scientific Publishing Company.

Lavrenko, V., Schmill, M., Lawrie, D., Ogilvie, P., Jensen, D., & Allan, J. (2000), Mining of Concurrent Text and Time Series, *KDD-2000 Workshop on Text Mining*, 37-44.

Lemire, D. (2007), A Better Alternative to Piecewise Linear Time Series Segmentation, To Appear in *SIAM International Conference on Data Mining*.

Lent, B., Agrawal, R., & Srikant, R. (1997), Discovering Trends in Text Databases, *Third International Conference on Knowledge Discovery and Data Mining*, 227-230.

McCue, P. & Hunter, J. R. W. (2004), Multivariate Segmentation of Time Series Data, *Intelligent Data Analysis in Medicine and Pharmacology*.

Muthukrishnan, S., Shah, R., & Vitter, J. S. (2004), Mining Deviants in Time Series Data Streams, *Sixteenth International Conference on Scientific and Statistical Database Management*, 41-50.

Oliver, J. J., & Forbes, C. S. (1997), Bayesian approaches to segmenting a simple time series, *Monash Econometrics and Business Statistics Working Papers*, 14/97.

Pavlidis, T. & Horowitz, S. L. (1974), Segmentation of Plane Curves, *IEEE Transactions on Computers*, C-23(8), 863-870.

Pednault, E. (1991), Minimal-Length Encoding and Inductive Inference, *Knowledge Discovery in Databases*, G. Piatetsky-Shapiro and W. Frawley (Eds.), AAAI Press, 71-92.

Siy, H., Chundi, P., Rosenkrantz, D. J., & Subramaniam, M. (2007), Discovering Dynamic Developer Relationships from Software Version Histories by Time Series Segmentation, *Twenty-Third International Conference on Software Maintenance (ICSM)*.

Tseng, V. S., Chen, C., Chen, C., & Hong, T. (2006), Segmentation of Time Series by the Clustering and Genetic Algorithms, *ICDM 2006 Workshop on Foundation of Data Mining and Novel Techniques in High Dimensional Structural and Unstructured Data,* 443-447.

KEY TERMS

Optimum Segmentation: An optimum segmentation for a given time series, with a given constraint *m* on the number of segments, is a segmentation with a minimum segmentation error among all segmentations containing at most *m* segments.

Segment: A segment is a sequence of one or more consecutive points from a time series, represented as a value or a model fitted to the data in these time points.

Segmentation: A segmentation of a given time series is a representation of the time series as a sequence of segments.

Segmentation Error: The segmentation error of a given segmentation is a measure of the overall difference between the representation of the segments in the segmentation and the data points in the time series.

Segment Error: The segment error of a given segment is a measure of the difference between the segment representation and the data points in the segment.

Time Series: A time series is a sequence of data points that represent the state of a variable, possibly having a composite structure, over consecutive points in time.

Uniform Segmentation: A uniform segmentation of a given time series is a segmentation all of whose segments are of the same length.

Segmenting the Mature Travel Market with Data Mining Tools

Yawei Wang
Montclair State University, USA

Susan A. Weston
Montclair State University, USA

Li-Chun Lin
Montclair State University, USA

Soo Kim
Montclair State University, USA

INTRODUCTION

The graying of America is one of the most significant demographic changes to the present and future of the United States (Moisey & Bichis, 1999). As more baby boomers enter their 50s and 60s, the mature travel market becomes a fast-growing market segment and starts to attract attention from many tourism researchers and professionals. The significant increases in size and wealth of the older population make the mature travel market a strong component of the general travel market (Reece, 2004). Understanding the mature market as well as mature travelers' motivations are vital to the success of the travel industry (Brewer, Poffley, & Pederson, 1995; Hsu, Cai, & Wong, 2007).

Today's mature travel market can be generalized as being "different, diverse and demanding" (Harssel, 1994, p. 376). Faranda and Schmidt (1999) suggest that mature tourism marketers must recognize three critical components: the aging process comprehended from multiple disciplines, the acknowledged "heterogeneity and dynamic nature" of the mature market, and the "necessity for sound segmentation methods" (p. 24). It is not a simple task for marketers to fully understand the mature travel market.

In order to better understand and serve the diverse market, tourism professionals will have to use data mining (DM) tools and techniques to discover the hidden patterns and characteristics of the mature travel market. According to Pyo, Uysal, and Chang (2002), DM can be applied to many areas in tourism research. These areas include destination quality control and perceptions, environmental scanning and optimization, travel behavior, tourism forecasting, and market segmentation and positioning. Therefore, the purpose of this study is to review and analyze the segmentation methods reported in the literature during the past seven years on the mature travel market and to explore the application of DM tools regarding the segmentation of the mature travel market in the near future.

BACKGROUND

A diversity of segmentation variables have been documented in the literature of the mature travel market. The segmentation efforts usually focus on socio-demographic variables (e.g., age, gender, and employment status) and psychographic variables (e.g., motivations and constraints). Demographic and behavioral profiles are then developed and compared based on subgroups or segments, with the help of data analytical tools.

A Priori Segmentation

A priori segmentation approach has been widely used in tourism studies (Dolnicar, 2004). Most tourism researchers use geographic and demographic characteristics to analyze the market (Hsu & Lee, 2002). Socio-demographic variables, such as age, gender, and retirement, are already known before conducting statistical analyses. The number of segments is known and determined by pre-selected variables.

Age. Mature travelers' biological age is always used as a categorical variable to distinguish different subgroups within the mature travel market. Various terms are employed to describe different age subgroups, such as "age cycles," "first cycle" and "next cycle" (Fleischer & Pizam, 2002), and "young old," "old," and "very old" (Hong, Kim, & Lee, 1999). The age subgroups are found different regarding to psychographic, behavior, and other sociodemographic variables. For example, younger senior travelers (55-64) reported a conservation/protection attitude rather than the consumptive attitude reported by the older senior travelers. Older senior travelers were more likely to visit friends and relatives than younger senior travelers (Backman, Backman, & Silverberg, 1999). Hong, Kim, and Lee (1999) found from a study of three age subgroups: young-old (55-64), old (65-74), and very old (75+) that race, education, marital status, and economic factors determined the decision to travel (i.e., whether or not to travel), while age, health care expenditures, and household income were found significant in predicting tourism expenditure.

Age cycles were associated with the effect of the constraints on the number of vacation days (Fleischer and Pizam, 2002). In the first cycle (55-65), the number of vacation days was positively correlated with leisure time and household income; while in the next cycle (65+), declining incomes and deteriorating health were found to cause a decrease of the vacation length.

Different from the traditional categorization based on chronological or objective age, Muller and O'Cass (2001) focused on the subjective age of older adults. Measured in felt age and activities age, subjective age was recognized as a valuable segmentation tool. The young-at-heart seniors sought fun and enjoyment in life, traveled for physical stimulation and a sense of accomplishment, and had high expectations of their vacation trips. The subjectively older seniors, those less young at heart, were concerned about security and they tended to worry about having trouble with travel arrangements, getting hurt or being in danger, and becoming ill on vacation. They preferred traveling with a group of friends or with their family, which distinguished them from the young-at-heart seniors.

Gender. The travel behavior based on gender in the context of mature tourism research has been rarely studied (Lehto, O'Leary, & Lee, 2001). Lehto et al. (2001) identified significant differences between male and female mature travelers in terms of preferences for destination attributes and travel products and services. Women were more drawn to people-oriented activities than male travelers. Older female travelers were also found interested in long-haul travel in general; however, they had strong preferences for shorter duration trips. Issues, such as personal safety, package or guided tours, and availability of comprehensive tourist information, were highly important to female travelers. Older male travelers paid attention to the utility or functional aspects of a travel destination.

Employment Status. Blazey (1992) investigated the relationship between pre- and post-retirement status, as well as key issues related to older adult travel activity. Four separate analyses examined the relationship between retirement status and constraints to travel activity, use of various forms of travel information, travel characteristics, and participation in travel related activities. Retirees were likely to be constrained by health conditions, physical energy, perception of age, and disability.

A Posteriori (Data-Driven) Segmentation

As noted by Hsu and Lee (2002), a posteriori approach is most likely used based on psychographic variables, such as motivations, constraints, and perceived benefits. In contrast to priori segmentation, researchers have no prior knowledge of the desired travel market segments regarding travelers' psychographic characteristics. DM analytical tools are required to generate the market segments.

Gerontologists proposed that as people reached their mature stage of life, they became more preoccupied with self-utilization. Cleaver, Muller, Ruys, and Wei (1999) stressed the strategic usefulness of identifying travel-motive segments for tourism product development. Seven travel-motive segments were determined with factor analyses, namely, Nostalgics, Friendlies, Learners, Escapists, Thinkers, Status-Seekers, and Physicals.

You and O'Leary (1999) examined the diversity and heterogeneity of the older UK outbound travelers' market and segmented it based on travel push and pull factors. The older market was categorized into three distinct groups, namely, passive visitors, the enthusiastic go-getters, and the culture hounds. The three segments exhibit distinct differences in demographics as well as their destination participation patterns,

travel philosophies, trip frequencies and other travel characteristics.

MAIN FOCUS

A total of seven journal articles published since 2000 in six different sources (i.e., Activities, Adaptation & Aging, Journal of Hospitality & Leisure Marketing, Journal of Travel Research, Journal of Travel & Tourism Marketing, Journal of Vacation Marketing, and Tourism Management) were selected and included in this study. All the articles focused on segmenting the mature travel market from a data-driven approach. A review of the sample size, segmentation variables, DM methods used to segment the market, as well as the results was summarized in Table 1.

As shown in Table 1, most of the recent literature on segmentation of the mature travel market was data-driven. Variables such as travel motives and benefits were commonly used to segment the mature travel market. Traveler attributes (i.e., age, gender, marital status, income, companions) and travel frequency were applied to a priori segmentation approach. More studies employ a posteriori segmentation to depict the heterogeneity of the market.

Due to its interdisciplinary nature, cluster analysis was the single DM technique that has been widely used in the recent segmentation literature except in Kim, Wei, and Ruys' (2003) study, although there were a wide variety of grouping techniques (Wedel & Kamakura, 1998). The fundamental idea of cluster analysis is to divide a number of survey respondents into some subgroups according to a pre-defined and pre-determined criterion. The similarity of individuals within the subgroups and the dissimilarity between them are assumed to be achieved to their maximum (Dolnicar, 2002).

Factor analysis and/or discriminant analysis were often used to accompany cluster analysis; for example,

Table 1. Summary of the Mature Travel Market Segmentation Studies

Study	Sample	Segmentation Variables	DM Methods	Segmentation Results
Guinn & Vincent (2003)	50+ visitors to a Winter destination in Texas (n=154)	Monthly means frequency of activity participation	Hierarchical cluster analysis; Stepwise discriminant analysis	- high involvement - low involvement
Hsu & Lee (2002)	55+ motorcoach travelers (n=817)	Motorcoach tour selection attributes	Factor analysis; Cluster analysis; Stepwise discriminant analysis ANOVA, Chi-square	- dependents - sociables - independents
Kim, Wei, & Ruys (2003)	50+ Australian (n=200)	Traveler attributes (age, gender, marital status, income, companions)	Kohonen's self-organizing maps (SOM, an artificial neural network)	- active learner - relaxed family body - careful participant - elementary vacationer
Lehto, O'Leary, & Lee (2001)	50+ French travelers	Traveler attributes	Cluster analysis; ANOVA	- independent eco-tourists - enthusiastic female experiencers - budget conscious relaxation seekers
Littrell, Paige, & Song (2004)	50+ Caucasian travelers (n=146)	Travel activities	Principle component factor analysis; K-means cluster analysis; ANOVA	- active outdoor/cultural tourists - cultural tourists - moderate tourists
Sellick (2004)	50+ members of the National Seniors Association in Australia (n=986)	Motives; Benefits; Value & lifestyle	Factor analysis; Cluster analysis; Multiple discriminant analysis	- discovery and self-enhancement - enthusiastic connectors - reluctant travelers - nostalgic travelers
Shoemaker (2000)	Pennsylvania residents 55+ (n=234)	Motivation	ANOVA; *K*-means clustering; Discriminant analysis	- escape and learn group - the retirees - active storytellers

two articles employed all the three DM tools (Hsu & Lee, 2002; Sellick, 2004), two studies used discriminant analysis to supplement cluster analysis (Guinn & Vincent, 2003; Shoemaker, 2000), and only one study conducted factor analysis (Littrell, Paige, & Song, 2004). Factor analysis was usually used first to reduce and purify the dimensionality of the selected attributes, and therefore to enhance the ability to classify and generate distinct subgroups within the mature travel market. A stepwise discriminant analysis was often conducted as a supplementary tool to further identify distinguishing attributes between the subgroups (Hsu & Lee, 2002).

FUTURE TRENDS

One of the emerging trends of data mining in the field of mature tourism is the application of various segmentation tools to the mature tourism research, such as neural network modeling. As discussed by Kim, et al. (2003), Kohonen's self-organizing maps (SOM) is capable of combining vector quantization and projection and therefore providing a visual data map of the subgroups. A two-dimensional Kohonen network possesses a layer of strongly interrelated neurons (e.g., features and attributes), which enables the implementation of a nonlinear and multidimensional mapping. Kim, et al. employed Kohonen's SOM to segment the motivations and concerns of the travelers based on older adults' traveler attributes and characteristics. The deployment of the neural network models enhances the travel market professionals' understanding of changing behavior among tourists within and between the macrosegments (Bloom, 2005).

The second trend is the employment of a diversity of the combined DM tools, as demanded by the urgent need for combined segmentation variables, for example, a combination of psychographic and demographic variables. A behavioral profile of the male and female subgroups based on a wide array of motivations is more accurate and persuasive in predicting travel trends than a profile purely based on gender differences. Therefore, to develop a series of the combined DM tools is necessary to advance the research in segmenting the mature travel market.

Exploring DM tools to investigate non-traditional segmentation variables that are relevant to the nature of the mature travel market is the third trend. Two types of variables can contribute to an enhanced understanding of market segmentation. First of all, it is necessary to add gerontographic variables to the segmentation process (Shoemaker, 2000). The gerontographic factors can depict the picture of the mature travelers to its full extent. For example, Moschis (1996) developed a gerontographic life stage model, which classifies older adults into four segments based on the amount and type of aging events and health problems they have experienced. These groups (i.e., healthy indulgers, healthy hermits, ailing outgoers, and freail recluses) are hypothesized to possess different attitudes toward travel. Secondly, a close connection between destination and travel decision makes it critical to include destination attributes into the market segmentation process. After defining the position of a destination, a profile of the target market should be proposed and developed by considering the destination attractions (Pyo, et al., 2002).

CONCLUSION

A review of the most recent segmentation studies in mature tourism revealed the overwhelming usage of cluster analysis and the emergence of other segmentation tools, for example, neural network modeling. With the increasing magnitude and diversification of the older population, the segmentation of the mature travel market is a promising field to which many DM tools and techniques may be practiced and applied. The employment of various DM tools, in turn, will enhance the tourism researchers' ability to understand and segment the mature travel market.

REFERENCES

Backman, K. F., Backman, S. J., & Silverberg, K. E. (1999). An investigation into the psychographics of senior nature-based travellers. *Tourism Recreation Research, 24*, 13-22.

Blazey, M. A. (1992). Travel and retirement status. *Annals of Tourism Research, 19*, 771-783.

Bloom, J. Z. (2005). Market segmentation: A neural network application. *Annals of Tourism Research, 32*, 93-111.

Brewer, K. P., Poffley, J. K., & Pederson, E. B. (1995). Travel interests among special seniors: Continuing care retirement community residents. *Journal of Travel & Tourism Marketing, 4*(2), 93-98.

Cleaver, M., Muller, T. E., Ruys, H. F. M., & Wei, S. (1999). Tourism product development for the senior market, based on travel-motive research. *Tourism Recreation Research, 24*, 5-11.

Dolnicar, S. (2002). A review of data-driven market segmentation in tourism. *Journal of Travel & Tourism Marketing, 12*(1), 1-22.

Dolnicar, S. (2004). Beyond "commonsense segmentation": A systematics of segmentation approaches in tourism. *Journal of Travel Research, 42*(3), 244-250.

Faranda, W. T. & Schmidt, S. L. (1999). Segmentation and the senior traveler: Implications for today's and tomorrow's aging consumer. *Journal of Travel and Tourism Marketing, 8*, 3-27.

Fleischer, A., & Pizam, A. (2002). Tourism constraints among Israeli seniors. *Annals of Tourism Research, 29*(1), 106-123.

Guinn, B, & Vincent, V. (2003). Activity participation among extended-stay senior travelers. *Activities, Adaptation & Aging, 27*(3/4), 39-51.

Harssel, J. V. (1994). The senior travel market: distinct, diverse, demanding. In W. F. Theobald (Eds.) *Global tourism: the next decade*. Oxford: Butterworth-Heinemann Ltd.

Hong, G. S., Kim, S. Y., & Lee, J. (1999). Travel expenditure pattern of elderly households in the U.S. *Tourism Recreation Research, 24*, 43-52.

Hsu, C. H. C., Cai, L. A., & Wong, K. K. F. (2007). A model of senior tourism motivations – Anecdotes from Beijing and Shanghai. *Tourism Management, 28*, 1262-1273.

Hsu, C. H. C., & Lee, E-J. (2002). Segmentation of senior motorcoach travelers. *Journal of Travel Research, 40*, 364-373.

Kim, J., Wei, S., & Ruys, H. (2003). Segmenting the market of West Australian senior tourists using an artificial neural network. *Tourism Management, 24*, 25-34.

Lehto, X. Y., O'Leary, J. T., & Lee, G. (2001). Mature international travelers: An examination of gender and benefits. *Journal of Hospitality & Leisure Marketing*, 9(1/2), 53-72.

Littrell, M. A., Paige, R. C., & Song, K. (2004). Senior travellers : Tourism activities and shopping behaviours. *Journal of Vacation Marketing, 10*, 348-362.

Moisey, R. N. & Bichis, M. (1999). Psychographics of senior nature tourists: The Katy nature trail. *Tourism Recreation Research, 24*, 69-76.

Muller, T. E., & O'Cass, A. (2001). Targeting the young at heart: Seeing senior vacationers the way they see themselves. *Journal of Vacation Marketing*, 7(4), 285-301.

Pyo, S., Uysal, M., & Chang, H. (2002). Knowledge Discovery in database for tourist destinations. *Journal of Travel Research, 40*, 396-403.

Reece, W. S. (2004). Are senior leisure travelers different? *Journal of Travel Research*, 43, 11-18.

Sellick, M. C. (2004). Discovery, connection, nostalgia: Key travel motives within the senior market. *Journal of Travel & Tourism Marketing, 17*(1), 55-71.

Shoemaker, S. (2000). Segmenting the mature market: 10 years later. *Journal of Travel Research*, 39, 11-26.

Wedel, M., & Kamakura, W. (1998) *Market segmentation–Conceptual and methodological foundations*. Boston, MA: Kluwer Academic Publishers.

You, X., & O'Leary, J. T. (1999). Destination behaviour of older UK travellers. *Tourism Recreation Research*, 24, 23-34.

KEY TERMS

A Posteriori Segmentation (Data-Driven or Post Hoc Segmentation): Quantitative techniques of data analysis are used to derive a grouping based on psychographic and behavioral segmentation variables.

A Priori Segmentation: A conceptual approach when the relevant dimensions for grouping respondents in an empirical study are felt to be known in advance, except for the fact that both uni- and multidimensional approaches are used (Dolnicar, 2002).

Baby Boomers: People who were born between 1946 to 1964 in the United States, as well as in Australia, Canada, and United Kingdom.

Data Mining (DM): A "discovery-oriented data analysis technology, which automatically detects hidden important information in the data warehouse" (Pyo, et al., 2002, p.397).

Market Segmentation: The process of dividing a total marketing into distinct subgroups, with shared characteristics within the subgroups.

Mature Travel Market: The travel market that is composed of older travelers, usually aged 50 and above.

Tourism: The activities of persons traveling to and staying in places outside their usual environment for not more than one consecutive year for leisure, business and other purposes (an official definition provided by the United Nations World Tourism Organization).

Travel Constraints: The factors that prevent people from traveling or from enjoying traveling (Fleischer & Pizam, 2002).

Travel Motivation: The reasons that make a person engaged in a tourism-related activity.

Semantic Data Mining

Protima Banerjee
Drexel University, USA

Xiaohua Hu
Drexel University, USA

Illhio Yoo
Drexel University, USA

INTRODUCTION

Over the past few decades, data mining has emerged as a field of research critical to understanding and assimilating the large stores of data accumulated by corporations, government agencies, and laboratories. Early on, mining algorithms and techniques were limited to relational data sets coming directly from On-Line Transaction Processing (OLTP) systems, or from a consolidated enterprise data warehouse. However, recent work has begun to extend the limits of data mining strategies to include "semi-structured data such as HTML and XML texts, symbolic sequences, ordered trees and relations represented by advanced logics." (Washio and Motoda, 2003)

The goal of any data mining endeavor is to detect and extract patterns in the data sets being examined. Semantic data mining is a novel approach that makes use of graph topology, one of the most fundamental and generic mathematical constructs, and semantic meaning, to scan semi-structured data for patterns. This technique has the potential to be especially powerful as graph data representation can capture so many types of semantic relationships. Current research efforts in this field are focused on utilizing graph-structured semantic information to derive complex and meaningful relationships in a wide variety of application areas-- national security and web mining being foremost among these.

In this article, we review significant segments of recent data mining research that feed into semantic data mining and describe some promising application areas.

BACKGROUND

In mathematics, a graph is viewed as a collection of vertices or nodes and a set of edges which connect pairs of those nodes; graphs may be partitioned into sub-graphs to expedite and/or simplify the mining process. A tree is defined as an acyclic sub-graph, and trees may be ordered or unordered, depending on whether or not the edges are labeled to specify precedence. If a sub-graph does not include any branches, it is called a path.

The two pioneering works in graph-based data mining, the algorithmic precursor to semantic data mining, take an approach based on greedy search. The first of these, SUBDUE, deals with conceptual graphs and is based on the Minimum Description Length (MDL) principle. (Cook and Holder, 1994) SUBDUE is designed to discover individual concepts within the graph by starting with a single vertex, which represents a potential concept, and then incrementally adding nodes to it. At each iteration, a more "abstract" concept is evaluated against the structure of the original graph, until the algorithm reaches a stopping point which is defined by the MDL heuristic. (Cook and Holder, 2000)

The second of the seminal graph mining works is called Graph Based Induction (GBI), and like SUBDUE, it is also designed to extract concepts from data sets. (Yoshida, Motoda, and Inokuchi, 1994) The GBI algorithm repeatedly compresses a graph by replacing each found sub-graph or concept with a single vertex. To avoid compressing the graph down to a single vertex, an empirical graph size definition is set to establish the size of the extracted patterns, as well as the size of the compressed graph.

Later researchers have applied several other approaches to the graph mining problem. Notable among these are the Apriori-based approach for finding frequent sub-graphs (Inokuchi, Washio, and Motoda, 2000; Kuramochi and Karypis, 2002), Inductive Logic Processing (ILP), which allows background knowledge to be incorporated in to the mining process; Inductive Database approaches which have the advantage of practical computational efficiency; and the Kernel Function approach, which uses the mathematical kernel function measure to compute similarity between two graphs. (Washio and Motoda, 2003)

Semantic data mining expands the scope of graph-based data mining from being primarily algorithmic, to include ontologies and other types of semantic information. These methods enhance the ability to systematically extract and/or construct domain specific features in data.

MAIN THRUST OF CHAPTER

Defining Semantics

The effectiveness of semantic data mining is predicated on the definition of a domain-specific structure that captures semantic meaning. Recent research suggests three possible methods of capturing this type of domain knowledge:

- Ontologies
- Semantic Associations
- Semantic Metadata

In this section, we will explore each of these in depth.

An ontology is a formal specification in a structured format, such as XML or RDF, of the concepts that exist within a given area of interest and the semantic relationships among those concepts. The most useful aspects of feature extraction and document classification, two fundamental data mining methods, are heavily dependent on semantic relationships. (Phillips and Buchanan, 2003) For example, a news document that describes "a car that ran into a gasoline station and exploded like a bomb" might not be classified as a terrorist act, while "a car bomb that exploded in a gasoline station" probably should be. (Gruenwald, McNutt and Mercier, 2003) Relational databases and flat documents alone do not have the required semantic knowledge to intelligently guide mining processes. While databases may store constraints between attributes, this is not the same as describing relationships among the attributes themselves. Ontologies are uniquely suited to characterize this semantic meta-knowledge. (Phillips and Buchanan, 2003)

In the past, ontologies have proved to be valuable in enhancing the document clustering process. (Hotho, Staab, and Strumme, 2003) While older methods of text clustering were only able to relate documents that used identical terminology, semantic clustering methods were able to take into account the conceptual similarity of terms such as might be defined in terminological resources or thesauri. Beneficial effects can be achieved for text document clustering by integrating an explicit conceptual account of terms found in ontologies such as WordNet. For example, documents containing the terms "beef" and "chicken" are found to be similar, because "beef" and "chicken" are both sub-concepts of "meat" and, at a higher level, "food". However, at a more granular clustering level, "beef" may be more similar to "pork" than "chicken" because both can be grouped together under the sub-heading of "red meat". (Hotho, Staab, and Strumme, 2003)

Ontologies have also been used to augment the knowledge discovery and knowledge sharing processes. (Phillips and Buchanan, 2003) While in the past, prior knowledge had been specified separately for each new problem, with the use of an ontology prior knowledge found to be useful for one problem area can be reused in another domain. Thus, shared knowledge can be stored even in a relatively simple ontology, and collections of ontologies can be consolidated together at later points in time to form a more comprehensive knowledge base.

At this point it should be noted that that the issues associated with ontology construction and maintenance are a research area in and of themselves. Some discussion of potential issues is presented in (Gruenwald, McNutt and Mercier, 2003) and (Phillips and Buchanan, 2003), but an extensive examination of this topic is beyond the scope of the current article.

In addition to ontologies, another important tool in extracting and understanding meaning is semantic associations. "Semantic associations lend meaning to information, making it understandable and actionable, and provide new and possibly unexpected insights." (Aleman-Meza, 2003) Looking at the Internet as a prime example, it becomes apparent that entities can be connected in multiple ways to other entities by types of relationships that cannot be known or established a priori. For example, a "student" can be related to a "university", "professors", "courses" and "grades" but she can also be related to other entities by different relations like financial ties, familial ties, neighborhood, etc. "In the Semantic Web vision, the RDF data model provides a mechanism to capture the meaning of an entity or resource by specifying how it relates to other entities or classes of resources" (Aleman-Meza, et. al., 2003) – each of these relationships between entities is a "semantic association" and users can formulate queries against them. For example, semantic association queries in the port security domain may include the following:

1. Are any passengers on a ship coming into dock in the United States known to be related by blood to one or more persons on the watch list?
2. Does the cargo on that ship contain any volatile or explosive materials, and are there any passengers on board that have specialized knowledge about the usage of those materials?

Semantic associations that span several entities and these constructs are very important in domains such as national security because they may enable analysts to uncover non-obvious connections between disparate people, places and events.

In conjunction with semantic associations, semantic metadata is an important tool in understanding the meaning of a document. Semantic metadata, in contrast to syntactic metadata, describes the content of a document, within the context of a particular domain of knowledge. For example, documents relating to the homeland security domain may include semantic metadata describing terrorist names, bombing locations, etc. (Sheth, et. al., 2002)

Methods of Graph Traversal

Once the semantic structures for a given domain have been defined, an effective method of for traversing those structures must be established. One such method that is coming into recent prominence, in addition to the algorithmic graph mining methods mentioned earlier in this chapter, is link mining. "Link mining is a newly emerging research area that is the intersection of the work in link analysis, hypertext and web mining, relational learning and inductive logic programming, and graph mining." (Getoor, 2003) Link mining is an instance of multi-relational data mining, in its broadest sense, and a field that is coming into prominence as the issues around graph traversal become paramount.

Link mining encompasses a range of tasks including both descriptive and predictive modeling. The field also introduces new algorithms for classification and clustering for the linked relational domain, and with the increasing prominence of links new mining tasks come to light as well. (Getoor, 2003) Examples of such new tasks include predicting the number of links between two entities, predicting link types, inferring the existence of a link based on existing entities and links, inferring the identity of an entity, finding co-references, and discovering sub-graph patterns. Link mining areas currently being explored are: link-based classification, which predicts the category of an object, link based cluster analysis, which clusters linked data based on the original work of SUBDUE, and several approaches on finding frequently occurring linking patterns. (Getoor, 2003).

Relative Importance and Ranking

There is increasing interest in developing algorithms and software tools for visualization, exploratory and interpretive analysis of graph-structured data, such the results of the semantic mining process. "While visualization techniques such as graph-drawing can be very useful for gleaning qualitative information about the structure of small graphs, there is also a need for quantitative tools for characterizing graph properties beyond simple lists of links and connections, particularly as graphs become too large and complex for manual analysis." (White and Smyth, 2003) In the area of web graphs, a number of ranking algorithms have been proposed, such as HITS (Kleinberg, 1999)

and PageRank (Brin and Page, 1998) for automatically determining the "importance" of Web pages.

One way of determining the relative importance of a result set might be to use a standard, global algorithm to rank all nodes in a sub-graph surrounding the root nodes of interest. The aforementioned PageRank algorithm is one such example. However, the problem with such an approach is that the root nodes are not given preferential treatment in the resulting ranking—in effect, one is ranking the nodes in the local sub-graph, rather than being ranked globally.

Another approach is to apply semantic methods themselves to the relevance and ranking problem. In order to determine the relevance of semantic associations to user queries, it becomes critical to capture the semantic context within which those queries are going to be interpreted and used, or, more specifically, the domains of user interest.

(Aleman-Meza, 2003) proposes that this can be accomplished "by allowing a user to browse an onotology and mark a region (sub-graph) of an RDF graph of nodes and/or properties of interest." The associations passing through these regions that are considered relevant are ranked more highly in the returned result set than other associations, which may be ranked lower or discarded.

FUTURE TRENDS

One of the most high-profile application areas for semantic data mining is in the building and mining of the Semantic Web, which associates the meaning of data with web content. The SCORE system (Semantic Content Organization and Retrieval Engine), built at the University of Georgia, is one example that uses semantic techniques to traverse relationships between entities. (Sheth, et. al., 2002) Designed as a semantic engine with main-memory based indexing, SCORE provides support for context sensitive search, browsing, correlation, normalization and content analysis. Once the semantic search engine determines the context of information described in the document, it can explore related entities through associations. By navigating these associations or relationships, the engine can access content about these entities.

Another critical domain for the application of semantic data mining, as mentioned previously, is national security. In this area, one of the most difficult aspects of the mining process is creating an ontology to be used for the duration of the task. "In classification of a terrorist incident we must identify violent acts, weapons, tactics, targets, groups and perhaps individuals." (Gruenwald, McNutt and Mercier, 2003) While many domains are topic-driven and focus on only a single classification area, the national security inherently requires a focused search across multiple topics in order to classify a document as terrorism related. Specifically, a document must be identified as being semantically related to multiple branches in a terrorism hierarchy to be positively marked as relevant to the national security domain. (Gruenwald, McNutt and Mercier, 2003)

The prototypical Passenger Identification, Screening and Threat Analysis Application (PISTA), developed at the University of Georgia is an example of the application of the semantic mining approach to the national security domain. "PISTA extracts relevant metadata from different information resources, including government watch-lists, flight databases, and historical passenger data." (Sheth, et. al., 2003) Using a semantic association based knowledge discovery engine, PISTA discovers suspicious patterns and classifies passengers into high-risk, low-risk and no-risk groups, potentially minimizing the burden of an analyst who would have to perform further investigation. While PISTA restricts its focus to flight security, a similar approach might be applied to other aspects of national security and terrorism deterrence, such as port security and bomb threat prediction.

One relatively novel area to which semantic mining techniques have recently been applied is in Money Laundering Crimes. (Zhang, Salerno, and Yu, 2003) Money laundering is considered a major federal offense, and with the development of the global economy and internet commerce, it is predicted that Money Laundering will become more prevalent and difficult to detect. The investigation of such crimes involves analyzing thousands of text documents in order to generate crime group models. These models group together a number of people or entities linked by certain attributes. These "attributes" typically are identified by the investiga-

tors based on their experiences and expertise, and consequently are very subjective and/or specific to a particular situation. (Zhang, Salerno and Yu, 2003) The resulting structure resembles a semantic graph, with the edges defined by the aforementioned attributes. Once this graphical model of the crime group model has been generated graph traversal and semantic query techniques may be used to automatically detect potential investigation scenarios.

CONCLUSION

Today, semantic data mining is a fast growing field due to the increasing interest in understanding and exploiting the entities, attributes, and relationships in graph-structured data, which occurs naturally in many fields. In this review article, we have presented a high-level overview of several important research areas that feed into semantic mining, as well as describing some prominent applications of specific techniques. Many accomplishments have been made in this field to date, however, there is still much work to be done. As more and more domains begin to realize the predictive power that can be harnessed by using semantic search and association methods, it is expected that semantic mining will become of the utmost importance in our endeavor to assimilate and make effective use of the ever-increasing data stores that abound in today's world.

REFERENCES

Brin, S. & Page, L. (1998). The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Proceedings of the 7th International World Wide Web Conference*: 107-117.

Aleman-Meza, B., Halascheck, C., Ismailcem, B., & Sheth, A. P. (2003). Context-Aware Semantic Association Ranking. SWDB 2003: 33-50.

Cook, D. & Holder, L. (1994). Substructure Discovery Using Minimum Description Length and Background Knowledge. *Journal of Artificial Intelligence Research,* 1: 231-255.

Cook, D. & Holder, L. (2000). *Graph-Based Data Mining. IEEE Intelligent Systems*, 15(2): 32-41.

Getoor, L. (2003). Link Mining: A New Data Mining Challenge. *ACM SIGKDD Explorations Newsletter*, 5(1): 5-9.

Gruenwald, L., McNutt, G. & Mercier, A. (2003). Using An Ontology to Improve Search in a Terrorism Database System. DEXA Workshops 2003: 753-757.

Hotho, A., Staab, S. & Stumme, G. (2003). Ontologies Improve Text Document Clustering. Third IEEE International Conference on Data Mining: 541-544.

Inokuchi, A., Washio, T. & Motoda, H. (2000). An Apriori-based Algorithm for Mining Frequent Substructure from Graph Data. *Proceedings of the 4th European Conference on Principles of Knowledge Discovery and Data Mining*, 1910: 13-23.

Kleinberg, J. (1999). Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM*, 46(5): 604-632.

Kuramochi, M. & Karypis, G. (2002). Mining Scientific Data Sets Using Graphs. *NSF Next Generation Data Mining Workshop*: 170-179.

Phillips, J. & Buchanan, B. G. (2001). Ontology-guided Knowledge Discovery in Databases. *International Conference on Knowledge Capture*: 123-130.

Sheth, A., Bertram, C., Avant, D., Hammond, B., Kochut, K. & Warke, Y. (2002), Managing Semantic Content for the Web, *IEEE Internet Computing*, July/August 2002: 80-87.

Sheth, A., Aleman-Meza, B., Arpinar, B., Bertram, C., Warke, Y., Ramakrishnan, C., Halaschek, C., Anyanwu, K., Avant, D., Arpinar, S. & Kochut, K. (2003). Semantic Association Identification and Knowledge Discovery for National Security Applications. *Technical Memorandum #03-009 of the LSDIS, University of Georgia*.

Washio, T. & Motoda, H. (2003) State of the Art of Graph-based Data Mining, *SIGKDD Explorations Special Issue on Multi-Relational Data Mining*, 5(1): 45-52.

White, S. & Smyth P. (2003). Algorithms for Estimating Relative Importance in Networks. *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*: 266-275.

Yoshida, K., Motoda, H., & Indurkhya, N. (1994). Graph Based Induction as a Unified Learning Framework. *Journal of Applied Intelligence*: 4: 297-328.

Zhang, Z., Salerno, J., & Yu, P. (2003). Applying Data Mining in Investigating Money Laundering Crimes. *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*: 747-752.

KEY TERMS

Graph: In mathematics, a set of vertices or nodes which are connect by links or edges. A pair of vertices that are connected by multiple edges yield a multi-graph; vertices that are connected to themselves via looping edge yield a pseudo-graph.

Graph-Based Data Mining: A method of data mining which is used to find novel, useful, and understandable patterns in graph representations of data.

Link Mining: A method of data mining that combines techniques from link analysis, hypertext and web mining, relational learning and inductive logic programming, and graph mining. Link mining places primary emphasis on links, and is used in both predictive and descriptive modeling.

Ontology: A formal specification in a structured format, such as XML or RDF, of the concepts that exist within a given area of interest and the semantic relationships among those concepts.

Semantic Associations: "The associations that lend meaning to information, making it understandable and actionable, and providing new and possibly unexpected insights" (Aleman-Meza, 2003)

Semantic Context: The specification of the concepts particular to a domain that help to determine the interpretation of a document.

Semantic Data Mining: A method of data mining which is used to find novel, useful, and understandable patterns in data, and incorporates semantic information from a field into the mining process.

Semantic Metadata: Metadata that describes the content of a document, within the context of a particular domain of knowledge. For example, for documents relating to the homeland security domain, semantic metadata may include terrorist names, group affiliations, etc.

Semantic Web: An extension of the current World Wide Web, proposed by Tim Berners-Lee, in which information is given a well-defined meaning. The Semantic Web would allow software agents, as well as humans, to access and process information content.

Syntactic Metadata: Metadata that describes a document's structure and/or format. For example, document language, document size, and MIME type might all be included as elements of syntactic metadata.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 523-528, copyright 2005 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

Semantic Multimedia Content Retrieval and Filtering

Chrisa Tsinaraki
Technical University of Crete, Greece

Stavros Christodoulakis
Technical University of Crete, Greece

INTRODUCTION

Several consumer electronic devices that allow capturing digital multimedia content (like mp3 recorders, digital cameras, DVD camcorders, smart phones etc.) are available today. These devices have allowed both the amateur and the professional users to produce large volumes of digital multimedia material, which, together with the traditional media objects digitized recently (using scanners, audio and video digitization devices) form a huge distributed multimedia information source. The multimedia material that is available today is usually organized in independent multimedia information sources, developed on top of different software platforms.

The Internet, the emergence of advanced network infrastructures that allow for the fast, efficient and reliable transmission of multimedia content and the development of digital multimedia content services on top of them form an open multimedia consumption environment. In this environment, the users access the multimedia material either through computers or through cheap consumer electronic devices that allow the consumption and management of multimedia content. The users of such an open environment need to be able to access the services offered by the different vendors in a transparent way and to be able to compose the different atomic services (like, for example, multimedia content filtering) into new, composite ones. In order to fulfill this requirement, *interoperability* between the multimedia content services offered is necessary.

Interoperability is achieved, at the syntactic level, through the adoption of standards. At the semantic level, interoperability is achieved through the integration of domain knowledge expressed in the form of domain *ontologies*. An ontology is a logical theory accounting for the intended meaning of a formal vocabulary, i.e. its ontological commitment to a particular conceptualization of the world (Guarino, 1998).

The standard that dominates in multimedia content description is the *MPEG-7* (Salembier, 2001), formally known as *Multimedia Content Description Interface*. It supports multimedia content description from several points of view, including media information, creation information, structure, usage information, textual annotations, media semantics, and low-level visual and audio features. Since the MPEG-7 allows the structured description of the multimedia content semantics, rich and accurate semantic descriptions can be created and powerful *semantic retrieval and filtering* services can be built on top of them.

It has been shown, in our previous research (Tsinaraki, Fatourou and Christodoulakis, 2003), that domain ontologies capturing domain knowledge can be expressed using pure MPEG-7 constructs. This way, domain knowledge can be integrated in the MPEG-7 semantic descriptions. The domain knowledge is subsequently utilized for supporting semantic personalization, retrieval and filtering and has been shown to enhance the retrieval precision (Tsinaraki, Polydoros and Christodoulakis, 2007).

Although multimedia content description is now standardized through the adoption of the MPEG-7 and semantic multimedia content annotation is possible, multimedia content retrieval and filtering (especially *semantic* multimedia content retrieval and filtering), which form the basis of the multimedia content services, are far from being successfully standardized.

We focus in this chapter on MPEG-7 based semantic multimedia retrieval and filtering and we introduce the *MPEG-7 Query Language (MP7QL)* and its compatible user preference model, which aim to provide standardized support to such services. The rest of the chapter is structured as follows: In the *Background*

section we present the state of the art in MPEG-7 based multimedia content retrieval and filtering; In the *Main Focus* section we introduce the MP7QL query language and its compliant user preference model; In the *Future Trends* section we outline the future research directions in semantic multimedia content retrieval and filtering and we conclude in the *Conclusions* section.

BACKGROUND

In this section we present the state of the art in MPEG-7 based multimedia retrieval and filtering. It has been mentioned in the introduction that MPEG-7 allows the creation of rich multimedia content descriptions, based on the different aspects of the content. Powerful retrieval and filtering capabilities can be built on top of these descriptions.

Several research groups have been working on MPEG-7 based multimedia content retrieval and filtering, exploiting different features of the MPEG-7 descriptions. The systems offering MPEG-7 based multimedia content retrieval and filtering are classified in three categories:

1. Systems that exploit the textual annotations together with the media-related elements of the MPEG-7 descriptors for retrieval and filtering support (Graves and Lalmas, 2002; Rogers, Hunter and Kosovic, 2003; Tseng, Lin and Smith, 2004). Multimedia content filtering in these systems utilizes the MPEG-7 *Filtering and Search Preferences (FASP)*, which allow the users to specify their preferences regarding multimedia content retrieval and filtering.
2. Systems that exploit the *MPEG-7 Visual* (ISO/IEC, 2001a) and *Audio* (ISO/IEC, 2001a) *Descriptors* in order to support multimedia content retrieval based on the low-level MPEG-7 features (Eidenberger and Breiteneder, 2003; Bertini, del Bimbo and Nunziati 2006). These systems cannot be transparently integrated with the MPEG-7 FASP for multimedia content filtering, because the MPEG-7 FASP do not allow the expression of the user preferences regarding the MPEG-7 low-level visual and audio features.
3. Systems that exploit the semantic metadata descriptions formed according to the Semantic DS of the MPEG-7 *Multimedia Description Schemes (MDS)* (ISO/IEC, 2003a) for semantic multimedia retrieval and filtering (Hammiche, Lopez, Benbernou, Hacid and Vakali, 2006; Tsinaraki, Fatourou and Christodoulakis, 2003; Tsinaraki, Polydoros and Christodoulakis, 2007). These systems cannot be fully exploited using the MPEG-7 FASP for multimedia content filtering, because the MPEG-7 FASP allow only the keyword-based expression of the user preferences regarding the multimedia content semantics.

The major limitation of the above research efforts is that each of them exploits some of the features of the MPEG-7 multimedia content descriptions, but none of them provides a uniform and transparent MPEG-7 retrieval and filtering framework. The most important efforts in the direction of MPEG-7 based retrieval and filtering that transparently exploits all the features of the MPEG-7 descriptions are the following:

- The use of plain XQuery (Chamberlin et al., 2005) on top of an XML repository for MPEG-7 based multimedia content retrieval (Lee et al., 2003). This approach does not take into account the peculiarities of the MPEG-7 description elements. Thus, the different MPEG-7 metadata description elements cannot be fully exploited. This happens because both the MPEG-7 semantic model and the domain knowledge integrated in the semantic MPEG-7 descriptions are expressed in an involved way and cannot be successfully exploited if they are accessed in the same way with the textual and the media-related elements of the MPEG-7 metadata descriptions. The low-level visual and audio features also need special treatment. It is difficult for the average user to express, using plain XQuery, query conditions on the semantics and/or the low-level features and even more difficult to combine such query conditions with textual and media-related query conditions. Finally, XQuery does not support queries with preference values, which allow the users to state which query conditions are more important for them.
- The use of the existing MPEG-7 FASP in order to allow multimedia content filtering and retrieval. The limitations of this approach are the following:

- Several MPEG-7 description elements are not present in the MPEG-7 FASPs. The most important among these elements are the semantic elements and the low-level visual and audio features;
- The boolean operators (AND/OR/NOT), as well as the comparison operators (equals, greater, less) cannot be explicitly specified in the MPEG-7 FASPs.

These limitations do not allow the expression of queries for every aspect of the MPEG-7 descriptions. In addition, due to the lack of semantic support, they cannot support semantic multimedia content retrieval and filtering.

In order to overcome the limitations of the existing approaches, a language for querying MPEG-7 descriptions is needed, with clear, MPEG-7 specific semantics (instead of the generic semantics of the XQuery). In response to this need, the International Organization for Standardization (ISO) has recently issued the *MPEG-7 Query Format Requirements (MP7QF)* (ISO/IEC, 2006), in order to guide the MPEG-7 query format standardization. The MP7QL query language, which is presented in the next section, is a query language that fulfils the ISO MPEG-7 Query Format Requirements.

MAIN FOCUS

In this section we introduce the *MPEG-7 Query Language (MP7QL)* (Tsinaraki and Christodoulakis, 2007), a powerful query language that we have developed for querying MPEG-7 descriptions. The MP7QL has the MPEG-7 as data model and satisfies the MP7QF Requirements. It allows for querying every aspect of an MPEG-7 multimedia content description, including semantics, low-level visual features and media-related aspects. It also allows for the exploitation of domain knowledge encoded using pure MPEG-7 constructs. In addition, the MP7QL allows the explicit specification of both boolean operators and preference values in order to allow both the combination of the query conditions according to the user intentions and the expression of the importance of the individual conditions for the users.

The MP7QL queries may utilize the user preferences and the usage history as context, thus allowing for personalized multimedia content retrieval and filtering. The user preferences regarding multimedia content retrieval and filtering may be expressed either according to the MPEG-7 FASP model or according to the MP7QL FASP model, which has the standard MPEG-7 FASP model as a special case.

The MP7QL query output has the form of MPEG-7 documents where the query results are represented as parts of standard MPEG-7 collections, guaranteeing that the MP7QL language has the closure property (Date, 1995). This allows view definition support and the capability to store the results of the query language expressions as new MPEG-7 descriptions that can be reused by the query language in a recursive manner.

The MP7QL has been expressed using both XML Schema (Fallside, 2001) and OWL (McGuinness and van Harmelen, 2004) syntax. The implementation of an MP7QL query processor is in progress, on top of an XML native database accessed by XQuery. In the rest of this section we present the input and the output format of the MP7QL as well as the MP7QL compliant FASP model.

The Input Format of the MP7QL Query Language

The input format of the MP7QL allows expressing query conditions on every aspect of an MPEG-7 multimedia content description.

The fundamental MP7QL element is the MP7QL query. MP7QL allows the explicit specification of boolean operators and preference values, so that the users can express accurately how the retrieval criteria should be combined and which is the relevant importance of the different criteria. Three MP7QL query types have been defined:

- *Queries with explicit preference values*. The preference values are integers in the range [-100, 100], with default value 10. The type, the range and the default value of the preference values are compatible with the preference values used in the MPEG-7 FASP;
- *Queries with explicit boolean operators* (AND/OR/NOT);
- *Queries with explicit preference values and boolean operators*.

An MP7QL query has a SELECT-FROM-WHERE syntax and is formally described using the regular expression syntax of (1).

Q=[Select][From][Where][OrderBy][GroupBy] (1)

The *Select* element of an MP7QL query allows the specification of the elements and/or attributes of the MPEG-7 descriptions that will be returned in the query results. The *Select* element of an MP7QL query is formally described using the regular expression syntax of (2).

Select=Item [format][transformationRules] [maxItems][numOfPageItems] [page][timeLimit]* (2)

The *Item* elements of *Select* represent, in the form of XPath expressions, the elements and/or the attributes of the MPEG-7 descriptions that should be returned for each of the query results. The *format* attribute represents the URI of the file, where the structure of the output display format (that is, the format in which the query results will be displayed in the user's terminal device) is specified and has as default value the URI of the default query output format. The *transformationRules* attribute represents the URI of the XSL stylesheet (Kay, 2005) that should be applied in the standard MP7QL output in order to be presented according to a different format. The *maxItems* attribute represents the maximum number of the query results that will be returned to the user and has "unbounded" as default value. The *numOfPageItems* attribute represents the number of the query results that will be displayed in each result page and has 10 as default value. The *timeLimit* attribute represents the time limit in seconds until which the query must be replied and has 300 as default value. The *page* attribute specifies which result page should be returned to the user and has 1 as default value.

The *From* element of an MP7QL query allows the specification of the type of the MPEG-7 descriptions on which the query will be posed and is formally described using the regular expression syntax of (3).

*From =FromItem** (3)

The *FromItem* elements of *From* may take predefined string values that specify the type (i.e. ImageType, VideoType, SemanticEntityDefinition, Ontology etc.) of the MPEG-7 descriptions on which the query will be posed. Notice that the MP7QL allows, in addition to the queries on MPEG-7 descriptions, queries on the semantic entities that satisfy specific criteria (for example, "give me the players of the soccer team Barcelona") as well as queries on the constructs of domain ontologies expressed using MPEG-7 syntax (for example, "give me the subclasses of *SoccerPlayer*").

The *OrderBy* element of an MP7QL query allows the specification of the criteria for ordering the result set and is formally described using the regular expression syntax of (4).

*OrderBy=Criterion** (4)

The *Criterion* elements of *OrderBy* represent ordering criteria. A *Criterion* element is formally described using the regular expression syntax of (5).

Criterion=Item [priority][order] (5)

The *Item* element of *Criterion* represents, as an XPath expression, an element or an attribute of the MPEG-7 descriptions, on which the ordering will be based. The *priority* attribute represents the priority of the element/attribute in ordering and has 0 as default value. The *order* attribute represents the type (ascending or descending) of the ordering based on the current element/attribute and has "ascending" as default value.

The *GroupBy* element of an MP7QL query specifies, in the form of an XPath expression, the attribute or element that will be used for grouping the query results.

The *Where* element of an MP7QL query allows the expression of the query conditions set by the user. The structure of the *Where* element is different for the different types of the MP7QL queries. In particular:

1. The *Where* element of an MP7QL query with explicit preference values (WWhere) is formally described using the regular expression syntax of (6).

   *WWhere=(WQS pv)** (6)

   pv is an explicit preference value and *WQS* is a query specification with explicit preference values, which represents the query conditions set by the user. The query conditions may refer both

to the multimedia content descriptions and to the domain knowledge utilized in them.

2. The *Where* element of an MP7QL query with explicit boolean operators (BWhere) is formally described using the regular expression syntax of (7).

$$BWhere=BQS[NOT]\ ((AND|OR)\ BQS\ [NOT])^* \quad (7)$$

BQS is a query specification with explicit boolean operators.

3. The *Where* element of an MP7QL query with explicit preference values and boolean operators (BWWhere) is formally described using the regular expression syntax of (8).

$$BWWhere=BWQS\ pv\ ((AND|OR)\ BWQS\ pv)^* \quad (8)$$

BWQS is a query specification with explicitly specified preference values and boolean operators.

The query specifications have been designed to allow expressing conditions on every aspect of a multimedia object that has been described using MPEG-7, so that the MP7QL may be used for querying any MPEG-7 multimedia object description. Thus, every element of an MPEG-7 multimedia object description has a corresponding query specification element in the MP7QL query specifications. The corresponding query specification element is used to specify the conditions that should hold on the values of the MPEG-7 elements of the segments retrieved.

The string comparison operators (contains, notContains, equals, startsWith, endsWith and keywords) and the number comparison operators (equals, greaterThan, greaterThanOrEqual, lessThan, lessThanOrEqual and differentFrom) may be explicitly specified on the elements of the MP7QL query specifications.

The MP7QL provides *Variables*, in order to support joins on the conditions about the features of the MPEG-7 descriptions. From a syntactic point of view, a variable is an identifier that begins with the "$" character.

The expressive power of the MP7QL query language has been evaluated both against the general-purpose ISO MPEG-7 Query Format Requirements and in a domain-specific (soccer) complete application (Tsinaraki and Christodoulakis, 2007). The MP7QL has been found to cover the general-purpose requirements and to allow the expression of all the domain-specific semantic queries of the soccer application.

The Output Format of the MP7QL Query Language

The MP7QL query results are structured as MPEG-7 descriptions, guaranteeing the closure property of the MP7QL language. If the user wishes to view the query results structured in another way, he should specify the display format and the XSL stylesheet that should perform the transformation in his display device in the *Select* element of the MP7QL queries.

The MP7QL query output format organizes the query results in MPEG-7 descriptions where the query result sets are represented by MPEG-7 collections. The result items returned by an MP7QL query form an MPEG-7 *Mixed Collection*, ordered according to the ordering criteria provided by the user in the *OrderBy* element. If the user has specified a grouping criterion in the *GroupBy* element of the MP7QL query, every group of result items is represented by a mixed collection element that contains the group items ordered according to the ordering criteria. If no results are returned, an empty MPEG-7 collection is returned. Every result item is represented in the collection that contains the query results by a *MixedCollection* element comprised of: (a) *Concept* elements that represent the rank and the relevance of the current result item; (b) A URI reference to the MPEG-7 description of the current result item, which is represented by a *ContentRef* element if the query is about multimedia content descriptions and by a *ConceptRef* element if the query is about reusable semantic entities or ontologies expressed in MPEG-7 syntax; and (c) The element that contains the MPEG-7 description elements of the current result item that were returned according to the user selections in the *Select* part of the query. It is a *Content* element if the query is about multimedia content descriptions and a *Concept* element if the query is about reusable semantic entities or ontologies expressed in MPEG-7 syntax.

The MP7QL Compliant FASP Model

The MP7QL Filtering and Search Preferences (FASP) allow the users to specify their preferences regarding multimedia content consumption. We decided

to develop a new FASP model, because the existing MPEG-7 FASP model does not allow to fully exploit the MPEG-7 multimedia content description capabilities during multimedia content filtering and context-based retrieval. The major limitations of the MPEG-7 FASP model are the lack of boolean and comparison operators and the incapability of expressing conditions on several MPEG-7 description elements, including the semantic ones. This way, semantic services cannot be successfully provided.

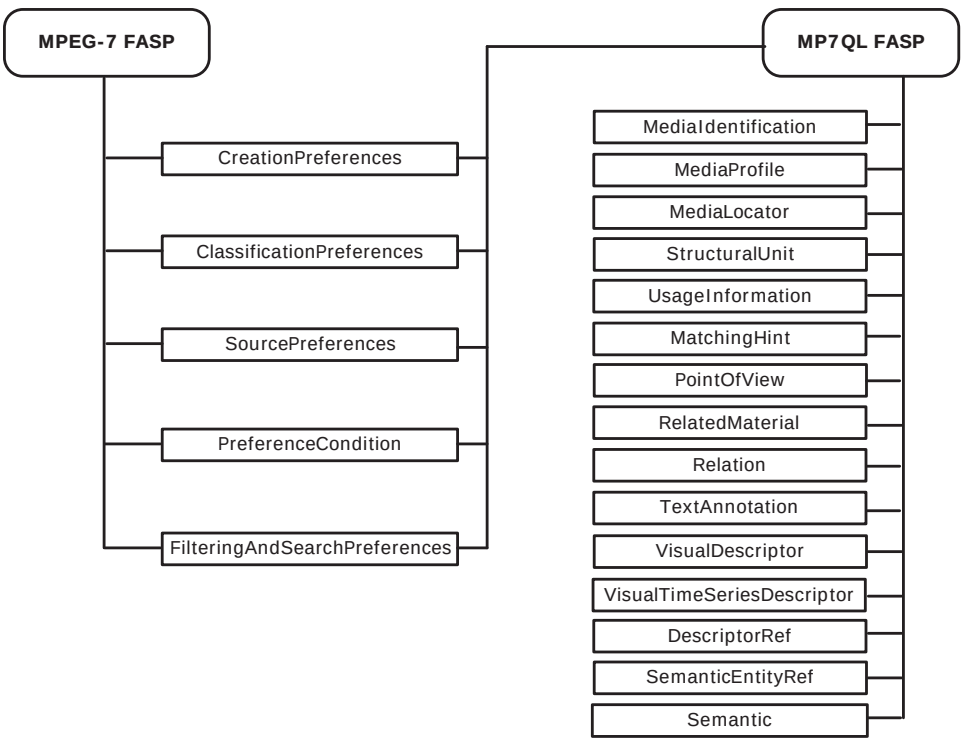
The MP7QL compliant FASP descriptions essentially are MP7QL query specifications. Thus, an MP7QL FASP may have all the elements of an MP7QL query specification. The MP7QL FASP are distinguished into FASP with explicit preference values (in the range [-100, 100]), FASP with explicit boolean operators and FASP with explicit boolean operators and explicit preference values. Notice that the NOT operator and the negative preference values allow the users to express their negative preferences (dislikes) in the FASP.

The MPEG-7 FASP model is a special case of the MP7QL FASP model. In particular, an MPEG-7 FASP is also an MP7QL FASP with explicit preference values, which has only the *CreationPreferences*, *ClassificationPreferences*, *SourcePreferences* , *FilteringAndSearchPreferences*, and *PreferenceCondition* elements (see Figure 1).

FUTURE TRENDS

The future trends in semantic multimedia retrieval and filtering include: (a) The standardization of the MPEG-7 query format. This will allow the standardization of multimedia query expression, in the same way that the MPEG-7 has allowed the standardization of multimedia content description; (b) The development of a retrieval and filtering model on top of the standard MPEG-7 query format, which will form the theoretical basis for satisfying the multimedia content retrieval and filtering requests; (c) The development of optimization techniques, which will allow the efficient MPEG-7 based multimedia retrieval and filtering; and (d) The specification of the MPEG-7 query server

Figure 1. The elements of the MPEG-7 FASP model and the MP7QL FASP model



capabilities description, using a profiling mechanism similar to the one that is available for MPEG-7 (ISO/IEC, 2003b). This way, the query servers provided by different vendors will be allowed to provide different, well-defined subsets of the standardized MPEG-7 query format functionality.

CONCLUSION

We have presented in this chapter the state of the art in semantic multimedia content retrieval and filtering and pointed out the limitations of the existing systems. Then, we introduced the MPEG-7 Query Language (MP7QL), a powerful query language that we have developed for querying MPEG-7 descriptions and its compatible FASP model. Finally, we presented the future trends in semantic multimedia retrieval and filtering.

REFERENCES

Bertini M., del Bimbo A. and Nunziati W. (2006). Video Clip Matching Using MPEG-7 Descriptors and Edit Distance. *Conference on Image and Video Retrieval (CIVR) 2006, pp. 133-142*.

Chamberlin D., Siméon J., Boag S., Fernández M., Florescu D. and Robie J., (eds.) (2005). XQuery 1.0: An XML Query Language. *W3C Recommendation* (2005). (http://www.w3.org/TR/xquery/).

Date C.J. (1995). An Introduction to Database Systems. *6th Edition, Addison-Wesley*, (1995).

Eidenberger H. and Breiteneder C. (2003). VizIR: A Framework for Visual Information Retrieval. *Journal of Visual Languages and Computing*, 14(5), 443-469 (2003).

Fallside D. (ed.) (2001). XML Schema Part 0: Primer. *W3C Recommendation*, 2001. (http://www.w3.org/TR/xmlschema-0/).

Graves A. and Lalmas M. (2002). Video Retrieval using an MPEG-7 based Inference Network. *ACM SIGIR*, pp. 339-346 (2002).

Guarino N. (1998). Formal Ontology and Information Systems. *1st International Conference "Formal Ontology in Information Systems" (FOIS '98), June 6-8 1998, pp. 3-15*.

McGuinness D. L. and van Harmelen F. (eds.) (2004). OWL Web Ontology Language: Overview. *W3C Recommendation, 2004*. (http://www.w3.org/TR/owl-features).

Hammiche S., Lopez B., Benbernou S., Hacid M.-S. and Vakali A. (2006). Domain Knowledge Based Queries for Multimedia Data Retrieval. *Workshop on Multimedia Semantics 2006 (WMS 2006), Chania, Crete, 2006, pp. 94-103*.

ISO/IEC (2001a). 15938-3:2001: Information Technology – Multimedia content description interface – Part 3 Visual. Version 1 (2001).

ISO/IEC (2001b). 15938-4:2001: Information Technology – Multimedia content description interface – Part 4 Audio. Version 1 (2001).

ISO/IEC (2003a). 15938-5:2003: Information Technology –Multimedia content description interface – Part 5: Multimedia description schemes. First Edition (2003).

ISO/IEC JTC1/SC29/WG11 (2006). N8219 – MPEG-7 Query Format Requirements version 1.1. Klagenfurt, Austria, July 2006.

ISO/IEC JTC1/SC29/WG11 (2003b). N6263 – Study of MPEG-7 Profiles Part 9. *Committee Draft*, December 2003.

Kay M. (ed.) (2005). XSL Transformations (XSLT) Version 2.0. *W3C Candidate Recommenda-tion*, 3 Nov. 2005. (http://www.w3.org/TR/xslt20/)

Lee M-H, Kang J-H, Myaeng S-H, Hyun S-J, Yoo J-M, Ko E-J, Jang J-W, Lim J-H (2003). A Multimedia Digital Library System based on MPEG-7 and XQuery. *6th International Conference on Asian Digital Libraries (ICADL), Kuala Lumpur, Malaysia, 2003, 193-205*.

Rogers D., Hunter J. and Kosovic D. (2003). The TV-Trawler Project. *International Journal of Imaging Systems and Technology*, 13(5), 289-296 (2003).

Salembier P. (2001). MPEG-7 Multimedia Description Schemes. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6), 748-759, (2001).

Tseng B., Lin C.-Y. and Smith J. (2004). Using MPEG-7 and MPEG-21 for personalizing video. IEEE Multimedia, 11(1), 42-52, 2004.

Tsinaraki C., Fatourou E. and Christodoulakis S. (2003). An Ontology-Driven Framework for the Management of Semantic Metadata describing Audiovisual Information. *Conference of Advanced Information Systems Engineering (CaiSE), Velden, Austria, 2003, pp 340-356*.

Tsinaraki C., Polydoros P. and Christodoulakis S. (2007). Interoperability support between MPEG-7/21 and OWL in DS-MIRF. *IEEE Transactions on Knowledge and Data Engineering*, 19(2), 219-232 (2007).

Tsinaraki C. and Christodoulakis S. (2007). An MPEG-7 Query Language and a User Preference Model that allow Semantic Retrieval and Filtering of Multimedia Content. *ACM-Springer Multimedia Systems Journal*, Special Issue on Semantic Multimedia Adaptation and Personalization, 2007.

KEY TERMS

Interoperability: The capability of different systems and services, possibly developed in different platforms and/or offered by different vendors to correctly work together, interact and make use of each other in an open environment.

MPEG-7: The MPEG-7, formally known as the Multimedia Content Description Interface, is the dominant standard in multimedia content description that allows the description of (segments of) multimedia objects from several points of view, including media information, creation information, structure, usage information, textual annotations, media semantics, and low-level visual and audio features.

MPEG-7 Filtering and Search Preferences (FASP): The part of the MPEG-7 user preferences that allows the users to specify their preferences regarding multimedia content retrieval and filtering

Ontology: A logical theory accounting for the intended meaning of a formal vocabulary, i.e. its ontological commitment to a particular conceptualization of the world.

Semantic Multimedia Filtering: Multimedia content filtering based on the user preferences on the semantics of the multimedia content.

Semantic Multimedia Retrieval: Multimedia content retrieval based on the semantics of the multimedia content.

User Preferences: Description of the preferences of the user regarding the consumption of multimedia content and the adaptation of the services accessed. The user preferences may be explicitly set by the user or computed based on his/her background and/or usage history.

Semi-Structured Document Classification

Ludovic Denoyer
University of Paris VI, France

Patrick Gallinari
University of Paris VI, France

INTRODUCTION

Document classification developed over the last ten years, using techniques originating from the pattern recognition and machine learning communities. All these methods do operate on flat text representations where word occurrences are considered independents. The recent paper (Sebastiani, 2002) gives a very good survey on textual document classification. With the development of structured textual and multimedia documents, and with the increasing importance of structured document formats like XML, the document nature is changing. Structured documents usually have a much richer representation than flat ones. They have a logical structure. They are often composed of heterogeneous information sources (e.g. text, image, video, metadata, etc). Another major change with structured documents is the possibility to access document elements or fragments. The development of classifiers for structured content is a new challenge for the machine learning and IR communities. A classifier for structured documents should be able to make use of the different content information sources present in an XML document and to classify both full documents and document parts. It should easily adapt to a variety of different sources (e.g. to different Document Type Definitions). It should be able to scale with large document collections.

BACKGROUND

Handling structured documents for different IR tasks is a new domain which has recently attracted an increasing attention. Most of the work in this new area has concentrated on ad hoc retrieval. Recent Sigir workshops (2000, 2002 and 2004) and journal issues (Baeza-Yates et al., 2002; Campos et. al., 2004) where dedicated to this subject. Most teams involved in this research gather around the recent initiative for the development and the evaluation of XML IR systems (INEX) which has been launched in 2002. Besides this mainstream of research, some work is also developing around other generic IR problems like clustering and classification for structured documents. Clustering has mainly been dealt with in the database community, focusing on structure clustering and ignoring the document content (Termier et al., 2002; Zaki and Aggarwal, 2003). Structured document classification the focus of this paper is discussed in greater length below.

Most papers dealing with structured documents classification propose to combine flat text classifiers operating on distinct document elements in order to classify the whole document. This has mainly been developed for the categorization of HTML pages. (Yang et al., 2002) combine three classifiers operating respectively on the textual information of a page, on titles and hyperlinks. (Cline, 1999) maps a structured document onto a fixed-size vector where each structural entity (title, links, text etc...) is encoded into a specific part of the vector. (Dumais and Chen, 2000) make use of the HTML tags information to select the most relevant part of each document. (Chakrabarti et al., 1998) use the information contained in neighboring documents of an HTML pages. All these methods explicitly rely on the HTML tag semantic, i.e.. they need to "know" whether tags correspond to a title, a link, a reference, etc. They cannot adapt to more general structured categorization tasks. Most models rely on a vectorial description of the document and do not offer a natural way for dealing with document fragments. Our model is not dependent of the semantic of the tags and is able to learn which parts of a document are relevant for the classification task.

A second family of models uses more principled approaches for structured documents. (Yi and Sundaresan, 2000) develop a probabilistic model for tree

like document classification. This model makes use of local word frequencies specific of each node so that it faces a very severe estimation problem for these local probabilities. (Diligenti et al., 2001) propose the Hidden Tree Markov Model (HTMM) which is an extension of HMMs to tree like structures. They performed tests on the WebKB collection showing a slight improvement over Naive Bayes (1%). Outside the field of Information Retrieval, some related models have also been proposed. The hierarchical HMM (Fine et al., 1998) (HHMM) is a generalization of HMMs where hidden nodes emit sequences instead of symbols for classical HMMs. The HHMM is aimed at discovering sub-structures in sequences instead of processing structured data.

Generative models have been used for flat document classification and clustering for a long time. Naive Bayes (Lewis, 1998) is one of the most used text classifier and different extensions have been proposed, e.g. (Koller and Sahami, 1997). Probabilistic models with latent variables have been used recently for text clustering, classification or mapping by different authors. (Vinokourov and Girolami, 2001; Cai and Hofmann, 2003). (Blei and Jordan, 2003) describe similar models for learning the correspondence between images or image regions and image captions. All these models do not handle structured representations.

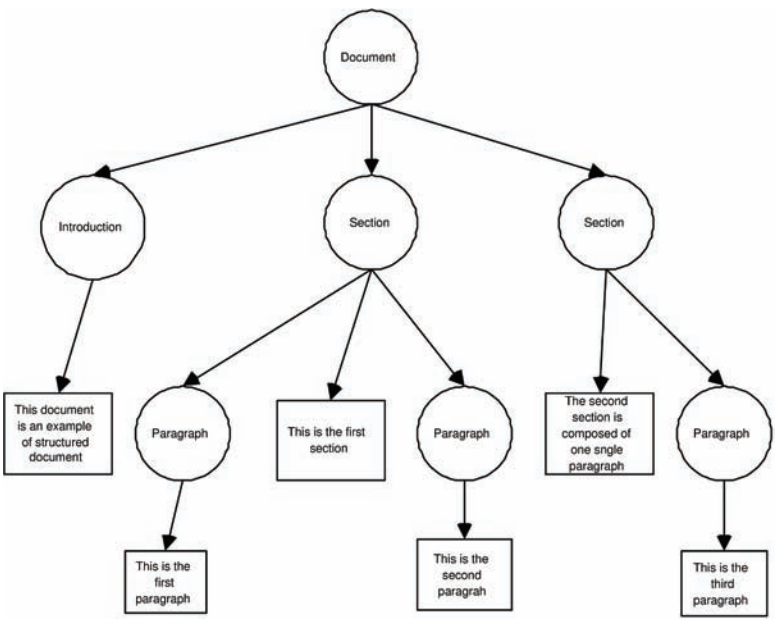
Finally, Bayesian networks have been used for the task of ad-hoc retrieval both for flat documents (Callan et al., 1992) and for structured documents (Myaeng et al., 1998; Piwowarski et al., 2002). This is different from classification since the information need is not specified in advance. The models and problems are therefore different from those discussed here.

MAIN THRUST

We describe a generative model for the classification of structured documents. Each document will be modeled by a Bayesian network. Classification will then amount to perform inference in this network. The model is able to take into account the structure of the document and different types of content information. It also allows one to perform inference either on whole documents or on document parts taken in their context, which goes beyond the capabilities of classical classifier schemes. The elements we consider are defined by the logical structure of the document. They typically correspond to the different components of an XML document.

In this chapter, we introduce structured documents and the core Bayesian network model. We then briefly summarize some experimental results and describe possible extensions of the model.

Figure 1. A tree representation for a structured document composed of an introduction and two sections. Circle and Square nodes are respectively structural and content nodes.



Structured Document

We will consider that a document is a tree where each node represents a structural entity. This corresponds to the usual representation of XML document. A node will contain two types of information:

- A label information which represents the type of the structural entity. A label could be for example *paragraph, section, introduction, title...* Labels depend on the documents corpora, for XML documents, they are usually defined in the DTD.
- A content information: for a multimedia document this could be text, image or signal. For a textual document node with label *paragraph*, the node content will be the paragraph text.

We will refer to structural and content nodes for these two types of information. Figure 1 gives an example for a simple textual document.

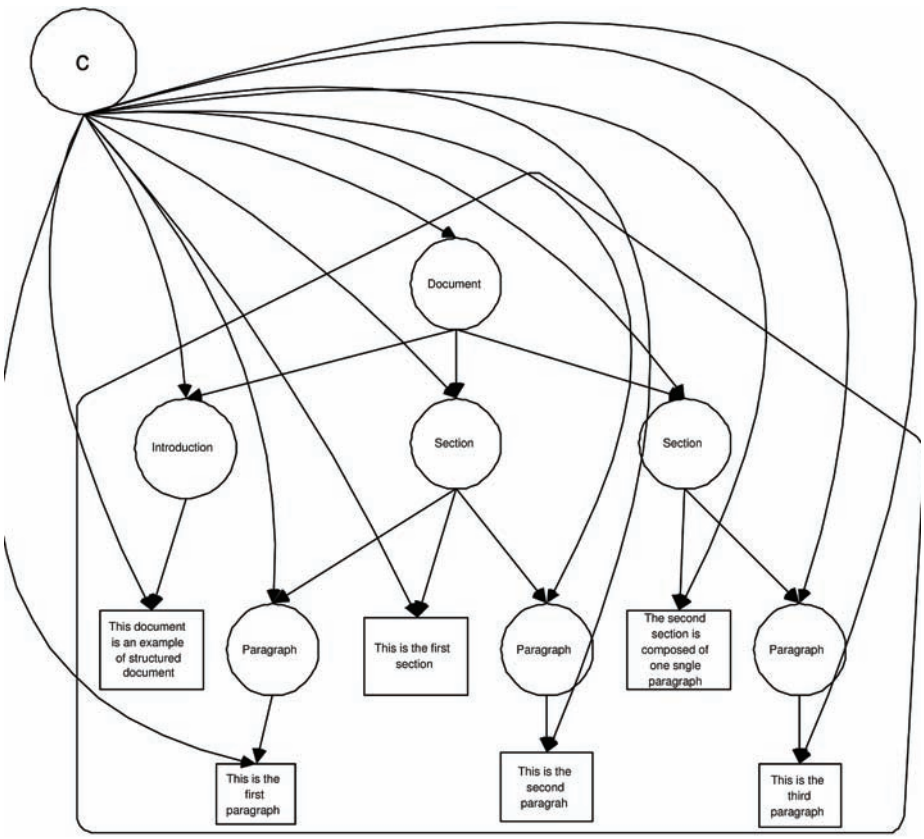
We will consider only textual documents here. Extensions for multimedia document are considered in (Denoyer et al., 2004a).

Modelling Documents with Bayesian Networks

Let us first introduce some notations:

- Let C be a discrete random variable which represents a class from the set of classes C.
- Let Λ be the set of all the possible labels for a structural node.
- Let V be the set of all possible words. denotes the set of all possible word sequences, including the empty one.
- Let d be a structured document consisting of a set of features where is the label of the i-th structural node of d (), is the textual content of this i-th node () and $|d|$ is the number of structural

Figure 2. A final bayesian network encoding «is a descendant of» relation



nodes. *d* is a realization of a random vector *D*. In the following, all nodes are supposed to have a unique identifier, indicated as superscript i.

Bayesian networks offer a suitable framework for modelling the dependencies and relations between the different elements in a structured document. We will associate a network model to each document. Since we focus here on the logical document structure, each network will be defined according to the corresponding document structure. For our classification task, the network parameters will be learned on all the documents from the same class in the training set. Documents from the same class will then share their parameters and there is one set of such parameters for each class.

Different networks could be used for modelling a document, depending on which type of relation we want to take into account. We consider here the explicit document structure and we will not try to uncover any hidden structure between the document elements. Some of the natural relations which could then be modelled are: "is a descendant of" in the document tree, "is a sibling of", "is a successor of" -given a preorder visit of the document tree-, and combinations of these different possibilities. Tests we performed using different types of relations and models of different complexity did not show a clear superiority of one model over the others with respect to classification performances. For simplifying the description, we will then consider tree like Bayesian networks. The network structure is built from the document tree, but need not be identical to this tree. Note that this is not limitative and all the derivations in the paper can be easily extended to BNs with no cycles. Figures 2 show a simple BN which encodes the "is a descendant of" relation and whose structure is similar to the document tree structure.

A Tree-Like model for Structured Document Classification

For this model, we make the following assumptions:

- There are two types of variables corresponding to structure and content nodes.
- Each structure node may have zero or many structure sub-nodes and zero or one content node.
- Each feature of the document depends on the class *c* we are interested in.
- Each structural variable depends on its parent in the document network.
- Each content variable depends only on its structural variable.

The generative process for the model corresponds to a recursive application of the following process: at each structural node *s*, one chooses a number of structural sub-nodes, which could be zero, and the length of the textual part if any. Sub-nodes labels and words are then sampled from their respective distribution which depends on *s* and the document class. The document depth could be another parameter of the model. Document length and depth distributions are omitted in our model since the corresponding terms fall out for the classification problems considered here.

Using such a network, we can write the joint content and structure probability:

$$P(d,c) = P(c)\underbrace{\left(\prod_{i=1}^{|d|} P(s_d^i \mid pa(s_d^i),c)\right)}_{(a)}\underbrace{\left(\prod_{i=1}^{|d|} P(t_d^i \mid s_d^i,c)\right)}_{(b)} \quad (1)$$

where (a) and (b) respectively correspond to **structural** and **textual probabilities.**

Structural probabilities can be directly estimated from data using some smooth estimator.

Since is defined on the infinite set , we shall make additional hypothesis for estimating the textual probabilities . In the following, we use a Naive Bayes model for text fragments. This is not a major option and other models could do as well. Let us define as the sequence of words where k and is the number of word occurrences i.e. the length of . Using Naive Bayes for the textual probability, the joint probability for this model is then:

$$P(d,c) = P(c)\left(\prod_{i=1}^{|d|} P(s_d^i \,|\, pa(s_d^i),c)\right)\left(\prod_{i=1}^{|d|}\prod_{k=1}^{|t_d^i|} P(w_{d,k}^i \,|\, s_d^i,c)\right) \quad (2)$$

Figure 3. The final document sub-net. In the full Bayesian network, all nodes also have node c for parent.

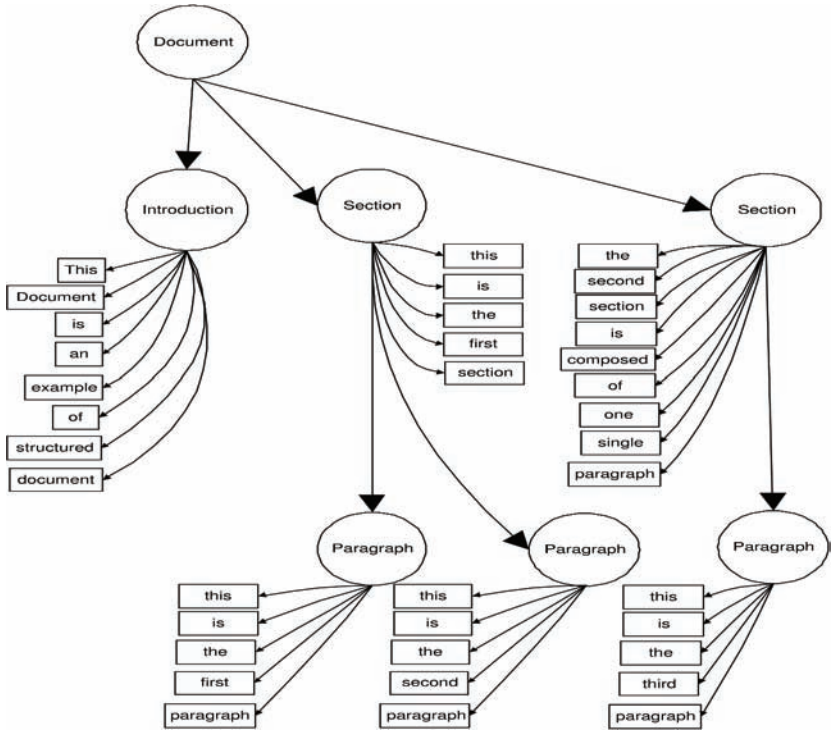


Figure 3 shows the final belief network obtained for the document in Figure 1. For clarity, the class variable is omitted.

Classifying Document Parts

Suppose now that *d* is a large heterogeneous document and that fragments of *d* correspond to different predefined classes. We could be interested into classifying any sub-part *d*' of *d* into one of these classes. If *d*' corresponds to a subtree of *d* and if we consider *d*' out of any context, we simply use Equation (2), replacing *d* with *d*'. We could also be interested into classifying *d*' within the context of document *d*. For this, we need to compute $P(d', c \mid d_{d'})$, where $d_{d'}$ represents *d* with *d*' removed. Let *s*' the structural node which is the father of *d*' root node. We get $P(d', c \mid d_{d'}) = P(d', c \mid s')$, which can be estimated via:

$$P(d', c \mid s') = P(c)\left(\prod_{i=k'}^{|d'|+k'} P(s_d^i \mid pa(s_d^i), c)\right)\left(\prod_{i=k'}^{|d'|+k'} \prod_{k=1}^{|t_d^i|} P(w_{d,k}^i \mid s_d^i, c)\right) \quad (3)$$

where *k*' is the index for the root of *d*' and structure nodes are supposed ordered according to a pre-order traversal of the tree. The interesting thing here is that by computing *P*(*d*,*c*), one automatically gets $P(d', c \mid d_{d'})$ since both quantities make use of the same probabilities and probability estimates. If *d*' does correspond to a partial sub-tree of *d* instead of a full sub-tree or to different subtrees in *d* one gets a similar expression by limiting the structure and content terms in the products in Equation (3) to those in *d*'. Classifying *d*' fragments is then easily performed with this generative classifier. This compositionality property (carrying out global computations by combining local ones) is achieved in this model via the probabilistic conditional independence assumptions. Compositionality is an important property for structured document classification. It is usually not shared by discriminant classifiers. Training discriminant classifiers both on document fragments might be prohibitive when the number of fragment is large e.g. the INEX corpus has about 16 K documents and 8 M fragments.

Learning

In order to estimate the joint probability of each document and each class, the parameters must be learned from a training set of documents. Let us define the θ parameters as:

$$\theta = \bigcup \left(\bigcup_{n\in\Lambda, m\in\Lambda} \theta_{n,m}^{c,s} \bigcup_{n\in V, m\in\Lambda} \theta_{n,m}^{c,w} \right) \quad (4)$$

where $_{s;n,m}$ is the estimation for and $_{w;n,m}$ is the estimation for . *s* in $_{s;.,.}$ indicates a structural parameter and *w* in $_{w;.,.}$ a textual parameter. There is one such set of parameter for each class.

For learning the θ*s* using the set of training documents *TRAIN*, we will maximize the log-likelihood *L* for *TRAIN* :

$$L = \sum_{d\in D_{TRAIN}} \log P(c) + \left(\sum_{i=1}^{|d|} \log \theta_{s_d^i, pa(s_d^i)}^{c,s} \right) + \left(\sum_{i=1}^{|d|} \sum_{k=1}^{t_d^i} \log \theta_{w_{d,k}, s_d^i}^{c,w} \right) \quad (5)$$

The learning algorithm solves for each parameter $_{.;n,m}$ (“.” corresponds to *s* or *w*) the following equation:

$_{.;n,m}$ under constraints :

$$\forall m \in \Lambda, \sum_{n\in\Lambda} \theta_{n,m}^{c,s} = 1$$
$$\forall m \in \Lambda, \sum_{n\in V} \theta_{n,m}^{c,w} = 1 \quad (6)$$

This equation has an analytical solution (Denoyer and Gallinari, 2004a).

In summary, this generative classifier can cope with both content and structure information. It allows one to perform inference on the different nodes and subtrees of the network. Document parts can then be classified in the context of the whole document. More generally decisions can be made by taking into account only a subpart of the document or when information is missing in the document.

(Denoyer and Gallinari, 2004a) describe how this model can take into account multimedia documents (text and image) and show how to extend it into a dicriminant classifier using the formalism of Fisher Kernels.

EXPERIMENTS

(Denoyer and Gallinari, 2004 a) describe experiments on three medium size corpus: INEX (about 15,000 scientic articles in XML,18 classes which correspond to journals), webKB (4520 HTML pages, 6 classes), NetProtect (19652 HTMLS pages with text and image, 2 classes). The BN model scales well on these corpus and outperforms Naïve Bayes with improvements ranging from 2% up to 6% (macro-average and micro-average recall) for whole document classification. These experiments validate experimentally the model and show the importance of taking into account both content and structure for classifying structured documents, even for the basic whole document classification tasks. The model also performs well for document fragment classification.

FUTURE TRENDS

We have presented a generative model for structured document. It is based on Bayesian networks and allows one to model the structure and the content of documents. Tests show that the model behaves well on a variety of situations. Further investigations are needed for analyzing its behavior on document fragments classification. The model could also be modified for learning implicit relations between document elements besides using the explicit structure so that the BN structure itself is learned. An interesting aspect of the generative model is that it could be used for other tasks relevant to IR. It could serve as a basis for clustering structured documents. The natural solution is to consider a mixture of Bayesian networks models where parameters do depend on the mixture component instead of the class as it is the case here. Two other important problems are Schema-mapping and automatic document structuring. These new tasks are currently being investigated in the database and IR communities. The potential of the model for performing inference on document parts when information is missing in the document will be helpful for this type of application. Preliminaries experiments about automatic structuration of documents are described in (Denoyer et al., 2004b).

REFERENCES

Baeza-Yates, R., Carmel, D., Maarek, Y., Soffer, A. (Eds.) (2002). *Journal of the American Society for Information Science and Technology* (JASIST).

Blei, D. M., Jordan, M. I. (2003). *Modeling annotated data*. In: Proceedings of SIGIR. ACM Press, 127–134.

Cai, L., Hofmann, T. (2003). *Text categorization by boosting automatically extracted concepts*. In: Proceedings of SIGIR. ACM Press, 182–189.

Callan, J. P., Croft, W. B., Harding, S. M. (1992). *The INQUERY retrieval system*. In: Proceedings of DEXA, 78–83.

Campos, L.M., Fernandez-Luna J.M., Huete J.F (Ed) (2004). *Information Processing & Management*, 40 (5).

Chakrabarti, S., Dom, B. E., Indyk, P.(1998). *Enhanced hypertext categorization using hyperlinks*. In: Haas, L. M., Tiwary, A. (Eds.) (1998). Proceedings of ACM-SIGMOD-98, 307–318.

Cline, M. (1999) *Utilizing HTML structure and linked pages to improve learning for text categorization*. Undergraduate Thesis, CS Dpt, University of Texas.

Denoyer, L. Gallinari P. (2004a). *Bayesian Network Model for Semi-Structured Document Classification*, in (Campos et al,

Denoyer, L., Wisniewski, G., Gallinari, P. (2004b). *Document Structure Matching for Heterogeneous Corpora*. In: SIGIR 2004, Workshop on Information Retrieval and XML. Sheffield, UK.

Diligenti, M., Gori, M., Maggini, M., Scarselli, F. (2001). *Classification of html documents by hidden tree-markov models*. Proceedings of ICDAR, 849–853.

Dumais, S. T., Chen, H. (2000). *Hierarchical classification of Web content*. In: Belkin, N. J., Ingwersen, P., Leong, M.-K. (Eds.) (2000). Proceedings of SIGIR-00. ACM Press, 256–263.

Fine, S., Singer, Y., Tishby, N. (1998). *The hierarchical hidden markov model: Analysis and applications*. Machine Learning, 32 (1), 41–62.

Fuhr, N., Govert, N., Kazai, G., Lalmas, M. (2002). *INEX: Initiative for the Evaluation of XML Retrieval*. Proceedings of ACM SIGIR 2002 Workshop on XML and Information Retrieval.

Koller, D., Sahami, M. (1997). *Hierarchically classifying documents using very few words*. In: Fisher, D. H. (Ed.), Proceedings of ICML, Morgan Kaufmann, 170-178.

Lewis, D. D., (1998). Naive (Bayes) at forty: *The independence assumption in information retrieval*. Proceedings of ECML-98,. Springer Verlag, , 4-15.

Myaeng, S. H., Jang, D.-H., Kim, M.-S., Zhoo, Z.-C. (1998). *A Flexible Model for Retrieval of SGML documents*. Proceedings of SIGIR. ACM Press, 138-140.

Piwowarski, B., Faure, G., Gallinari, P., Dec. (2002). *Bayesian networks and INEX*. Proceedings of the First Annual Workshop of the Initiative for the Evaluation of XML retrieval (INEX), Dagstuhl, Germany.

Sebastiani, F. (2002). *Machine learning in automated text categorization*. ACM Computing Surveys 34 (1), 1-47.

Termier, A., Rousset, M., Sebag, M. (2002). *Treefinder: a first step towards XML data mining*. In: ICDM. 450-457.

Vinokourov, A., Girolami, M. (2001). *Document classification employing the Fisher kernel derived from probabilistic hierarchic corpus representations*. Proceedings of ECIR-01. 24–40.

Yang, Y., Slattery, S., Ghani, R. (2002). *A study of approaches to hypertext categorization*. Journal of Intelligent Information Systems, 18 (2/3), 219–241.

Yi, J., Sundaresan, N. (2000). *A classifier for semi-structured documents*. Proceedings of the sixth ACM SIGKDD. ACM Press, 340–344.

Zaki, M. J., Aggarwal, C. C. (2003). *Xrules: An effective structural classifier for xml data*. SIGKDD 03. Washington, DC, USA.

KEY TERMS

XML (**Extensible Markup Language**): A W3C recommendation for creating special-purpose markup languages. It is a simplified subset of SGML, capable of describing many different kinds of data. Its primary

purpose is to facilitate the sharing of structured text and information across the Internet.

A Bayesian Network: A directed acyclic graph of nodes representing variables and arcs representing dependence relations among the variables.

Machine Learning: An area of artificial intelligence involving developing techniques to allow computers to "learn". More specifically, machine learning is a method for creating computer programs by the analysis of data sets, rather than the intuition of engineers.

Information Retrieval (IR): The art and science of searching for information in documents, searching for documents themselves, searching for metadata which describes documents, or searching within databases, whether relational stand alone databases or hypertext networked databases such as the Internet or intranets, for text, sound, images or data.

Multimedia: Data combining several different media, such as text, images, sound and video.

Probabilistic Model: A classic model of document retrieval based on a probabilistic interpretation of document relevance (to a given user query).

Semi-Structured Data: Data whose structure may not match, or only partially match, the structure prescribed by the data schema

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 1015-1021, copyright 2005 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

Semi-Supervised Learning

Tobias Scheffer
Humboldt-Universität zu Berlin, Germany

INTRODUCTION

For many classification problems, unlabeled training data are inexpensive and readily available, whereas labeling training data imposes costs. Semi-supervised classification algorithms aim at utilizing information contained in unlabeled data in addition to the (few) labeled data.

Semi-supervised (for an example, see Seeger, 2001) has a long tradition in statistics (Cooper & Freeman, 1970); much early work has focused on Bayesian discrimination of Gaussians. The Expectation Maximization (EM) algorithm (Dempster, Laird, & Rubin, 1977) is the most popular method for learning generative models from labeled and unlabeled data. Model-based, generative learning algorithms find model parameters (e.g., the parameters of a Gaussian mixture model) that best explain the available labeled and unlabeled data, and they derive the discriminating classification hypothesis from this model.

In discriminative learning, unlabeled data is typically incorporated via the integration of some model assumption into the discriminative framework (Miller & Uyar, 1997; Titterington, Smith, & Makov, 1985). The Transductive Support Vector Machine (Vapnik, 1998; Joachims, 1999) uses unlabeled data to identify a hyperplane that has a large distance not only from the labeled data but also from all unlabeled data. This identification results in a bias toward placing the hyperplane in regions of low density $p(x)$. Recently, studies have covered graph-based approaches that rely on the assumption that neighboring instances are more likely to belong to the same class than remote instances (Blum & Chawla, 2001).

A distinct approach to utilizing unlabeled data has been proposed by de Sa (1994), Yarowsky (1995) and Blum and Mitchell (1998). When the available attributes can be split into *independent* and *compatible* subsets, then *multi-view learning* algorithms can be employed. Multi-view algorithms, such as co-training (Blum & Mitchell, 1998) and co-EM (Nigam & Ghani, 2000), learn two independent hypotheses, which bootstrap by providing each other with labels for the unlabeled data.

An analysis of why training two independent hypotheses that provide each other with conjectured class labels for unlabeled data might be better than EM-like self-training has been provided by Dasgupta, Littman, and McAllester (2001) and has been simplified by Abney (2002). The disagreement rate of two independent hypotheses is an upper bound on the error rate of either hypothesis. Multi-view algorithms minimize the disagreement rate between the peer hypotheses (a situation that is most apparent for the algorithm of Collins & Singer, 1999) and thereby the error rate.

Semi-supervised learning is related to active learning. *Active learning* algorithms are able to actively query the class labels of unlabeled data. By contrast, semi-supervised algorithms are bound to learn from the given data.

BACKGROUND

Semi-supervised classification algorithms receive both labeled data $D_l = (x_1, y_1),..., (x_{m_l}, y_{m_l})$ and unlabeled data $D_u = x^u_1,..., x^u_{m_u}$ and return a classifier $f : x \rightarrow y$; the unlabeled data is generally assumed to be governed by an underlying distribution $p(x)$, and the labeled data by $p(x,y) = p(y \mid x)\ p(x)$. Typically, the goal is to find a classifierthat minimizes the error rate with respect to $p(x)$.

In the following sections, we distinguish between *model-based* approaches, mixtures of *model-based and discriminative* techniques, and *multi-view learning*. Model-based approaches can directly utilize unlabeled data to estimate $p(x, y)$ more accurately. Discriminative classification techniques need to be augmented with some model-based component to make effective use of unlabeled data. Multi-view learning can be applied

when the attributes can be split into two independent and compatible subsets.

Model-Based Semi-Supervised Classification

Model-based classification algorithms assume that the data be generated by a parametric mixture model $p(x, y \mid \Theta)$and that each mixture component contains only data belonging to a single class. Under this assumption, in principle, only one labeled example per mixture component is required (in addition to unlabeled data) to learn an accurate classifier. Estimating the parameter vector Θ from the data leads to a *generative* model; that is, the model $p(x, y \mid \Theta)$ can be used to draw new labeled data.

In the context of classification, the main purpose of the model is *discrimination.* Given the model parameter, the corresponding classifier is $f_\Theta(x) = \arg\max_y p(x, y \mid \Theta)$. For instance, when $p(x, y \mid \Theta)$ is a mixture of Gaussians with equal covariance matrices, then the discriminator $f_\Theta(x)$ is a linear function; in the general Gaussian case, $f_\Theta(x)$ is a second-order polynomial. The Expectation Maximization (EM) algorithm (Dempster et al., 1977) provides a general framework for semi-supervised model-based learning — that is, for finding model parameters Θ. Semi-supervised learning with EM is sketched in Table 1; after initializing the model by learning from the labeled data, it iterates two steps. In the E-step, the algorithm calculates the class probabilities for the unlabeled data based on the current model. In the M-step, the algorithm estimates a new set of model parameters from the labeled and the originally unlabeled data for which probabilistic labels have been estimated in the E-step.

The EM algorithm, which is a greedy method for maximizing the likelihood $p(D_l, D_u \mid \Theta) = p(D_l \mid \Theta)\, p(D_u \mid \Theta)$ of the data, has three caveats. The first is that no obvious connection exists between the *maximum likelihood* model parameters Θ and the Bayesian discriminator that minimizes the conditional risk given a new instance x. Practical semi-supervised learning algorithms apply some form of regularization to approximate the *maximum a posteriori* rather than the *maximum likelihood* parameters. The second caveat is that the resulting parameters are a *local* but not necessarily the *global* maximum. The third caveat of semi-supervised learning with EM is more subtle: When the assumed parametric model is *correct* — that is, the data has, in fact, been generated by $p(x, y \mid \Theta)$for some Θ— then the idea is arguable that unlabeled data will improve the accuracy of the resulting classifier $f_\Theta(x)$ under fairly reasonable assumptions (Zhang & Oles, 2000; Cozman, Cohen, & Cirelo, 2003). However, as Cozman et al. have pointed out, the situation is different when the model assumption is *incorrect* — that is, no Θ exists such that $p(x, y \mid \Theta)$ equals the true probability $p(x, y)$, which governs the data. In this case, the best approximation to the labeled data — $\Theta_l = \arg\max_\Theta p(D_l, \mid \Theta)$ — can be a much better classifier f_{Θ_l}than $f_\Theta(x)$ with $\Theta = \arg\max_\Theta p(D_l, D_u \mid \Theta)$, which approximates the labeled and unlabeled data. In other words, when the model assumption is incorrect, then semi-supervised learning with EM can generally result in poorer classifiers than supervised learning from only the labeled data.

Semi-supervised learning with EM has been employed with many underlying models and for many applications, including mixtures of Gaussians and naïve Bayesian text classification (Nigam, McCallum, Thrun, & Mitchell, 2000).

Table 1. Semi-supervised classification with EM.

Input: labeled data $D_l = (x_1, y_1), \ldots, (x_{m_l}, y_{m_l})$; unlabeled $D_u = x^u_1, \ldots, x^u_{m_u}$.
Initialize model parameters Θ by learning from the labeled data.
Repeat until a local optimum of the likelihood $p(x, y \mid \Theta)$ is reached.
E-step: For all unlabeled data x^u_i and class labels y, calculate
$E(f(x^u_1) = y \mid \Theta)$, the expected probability that y is the class of x^u_1 given Θ; that is, use $p(y \mid x, \Theta)$ to probabilistically label the x^u_i.
M-step: Calculate the maximum likelihood parameters
Θ = arg max $p(D_l, D_u$ estimated class probabilities for $Du)$; that is, learn from the labeled and probabilistically labeled unlabeled data.
Return classifier $p(y \mid x, \Theta)$.

Mixtures of Discriminative and Model-Based Learning

The answer to the question of how to utilize unlabeled data in the context of discriminative learning is not obvious. Discriminative learners, such as decision trees, logistic regression, or the Support Vector Machine, directly learn a classifier $y = f(x)$ without taking the detour via a generative model $p(x, y \mid \Theta)$. This classifier contains some information about the posterior $p(y \mid x)$ but does not contain a model of $p(x)$ that could be refined by unlabeled data. Some approaches that mix generative and discriminative models have been studied and use the unlabeled data to tweak their models of the class-conditional likelihood or the mixing probabilities (Miller & Uyar, 1997).

A special model assumption on the class-conditional likelihood $p(x \mid y)$underlies graph-based algorithms. It is often reasonable to assume that similar instances are more likely to have identical class labels than remote instances; clearly, the concept of similarity is very domain specific. In this situation, instances and similarities can be encoded in a graph, and mincuts of that graph correspond to optimal labelings of the unlabeled data (Blum & Chawla, 2001; Zhu, Gharamani, & Lafferty, 2003; Joachims, 2003).

The Transductive Support Vector Machine (Vapnik, 1998; Joachims, 1999) uses unlabeled data to identify a hyperplane that has a large distance not only from the labeled data but also from all unlabeled data. This identification is achieved by a greedy procedure that conjectures class labels for the unlabeled data and then iteratively flips the pair of conjectured class labels that yields the greatest improvement of the optimization criterion. After each flip of class labels, the hyperplane has to be retrained. This procedure results in a bias toward placing the hyperplane in regions of low density $p(x)$, where few instances result in small sums of the slack x_1 terms .

Multi-View Learning

Multi-view learning is a semi-supervised learning paradigm that is fundamentally different from model-based semi-supervised learning. Multi-view algorithms require the available attributes to be split into two independent subsets, or views, and either view has to be sufficient for learning the target concept. I discuss multi-view learning in the following section. The multi-view approach has also been applied to *active* learning (Muslea, Kloblock, & Minton, 2002).

MAIN THRUST

In this section, I discuss the multi-view framework of semi-supervised learning. In particular, I show that multi-view learning can, in a fundamentally different way, improve classification results over supervised learning, and I review some multi-view algorithms.

Multi-view learning applies when the available attributes can be decomposed into two views V_1 and V_2. For instance, V_1 can be the bag-of-words representation of a Web page, whereas V_2 might consist of the inbound hyperlinks referring to the page. Multi-view algorithms require that either view be *sufficient* for learning — that is, there exist functions f_1 and f_2 such that for all x, $f_1(x_1) = f_2(x_2) = f(x)$, where f is the true target function. This rule is also called the *compatibility* assumption. In addition, the views have to be conditionally independent given the class label — that is, $P(x_1, x_2 \mid y) = P(x_1 \mid y)\, P(x_2 \mid y)$.

In these independent views, independent classifiers f_1 and f_2 can be trained. Now Abney (2002) has observed the following: For an unlabeled instance x, you cannot decide whether $f_1(x)$ and $f_2(x)$ are correct or incorrect, but you can decide whether they agree or disagree. You can reasonably assume that either hypothesis has an error probability of no more than ½. For any given instance x with true class y, the probability of a disagreement is then an upper bound on the probability that either hypothesis misclassifies x. This can be shown by the following equations, which first utilize the independence, followed by the assumption that the error is at most ½, and finally the independence again.

This observation motivates the strategy that multi-view algorithms follow: minimize the error on the labeled data, and minimize the disagreement of the two

$$
\begin{aligned}
&P(f_1(x) \neq f_2(x)) \\
&= P(f_1(x) = y, f_2(x) = \overline{y}) + P(f_1(x) = \overline{y}, f_2(x) = y) \\
&\geq \max_i P(f_i(x) = y, f_i(x) = \overline{y}) + P(f_i(x) = \overline{y}, f_i(x) = y) \\
&= \max_i P(f_i(x) \neq y)
\end{aligned}
$$

independent hypotheses on the unlabeled data. Even though the error itself cannot be minimized on unlabeled data due to the absence of labels, by minimizing the disagreement on the unlabeled data, multi-view algorithms minimize an upper bound on the error. The most prominent learning algorithms that utilize this principle are co-training and co-EM, displayed in Table 2. Co-training can be wrapped around any learning algorithm with the ability to provide a confidence score for the classification of an instance. Here, two hypotheses bootstrap by providing each other with labels for the unlabeled examples that they are most confident about.

Co-EM, the multi-view counterpart of semi-supervised learning with EM, requires the base learner to be able to infer class label probabilities for the unlabeled data. In addition, a model has to be learned from conjectured class probabilities for the originally unlabeled data in addition to the labeled data. Because of these requirements, co-EM has frequently been applied with naïve Bayes as an underlying learner (for an example, see Nigam & Ghani, 1999). Recently, Brefeld and Scheffer (2004) studied a co-EM version of the Support Vector Machine; it transforms the uncalibrated decision function values into normalized probabilities and maps these probabilities to example-specific costs, which are used as weights for the error terms x_i in the optimization criterion. Co-EM is more effective than co-training, provided that the independence of the views is not violated (Muslea et al., 2002; Brefeld & Scheffer, 2004).

Multi-view algorithms require the views to be independent; this assumption will often be violated in practice. Muslea et al. (2002) has observed that co-EM, especially, is detrimental to the performance when dependence between attributes is introduced. Brefeld & Scheffer (2004) have observed co-training to be more robust against violations of the independence assumptions than co-EM; Krogel & Scheffer (2004) found even co-training to deteriorate the performance of SVMs for the prediction of gene functions and localizations. This discovery raises the questions of how dependence between views can be quantified and measured and which degree of dependence is tolerable for multi-view algorithms.

It is not possible to measure whether two large sets of continuous attributes are independent. The proof of Abney (2002) is based on the assumption that the two classifiers err independently; you can measure the violation of this assumption as follows Let E_1 and E_2 be two random variables that indicate whether f_1 and

Table 2. Semi-supervised classification with co-training and co-EM

Input: labeled data $D_l = (x_1, y_1),..., (x_{m_l}, y_{m_l})$; unlabeled data $D_u = x^u_1,..., x^u_{m_u}$, attributes are split into two views V_1 and V_2.
Algorithm Co-Training:
For $v = 1..2$: Learn initial classifier f_v from labeled data D_l in view v.
Repeat until D_u is empty.
  For $v = 1..2$: find the examples that f_v most confidently rates positive and negative, remove them from D_u and add them to D_l, labeled positive and negative, respectively.
For $v = 1..2$: learn new classifier f_v from labeled data D_l in view v .

Algorithm Co-EM:
Learn parameters Θ_2 of initial classifier $f_{(\Theta,2)}$ from labeled data D_l in view 2.
Repeat for T iterations.
  For $v = 1..2$:
    M-Step: Estimate class probabilities $p_v(y \mid x, \Theta_{\bar{v}})$ of unlabeled data using the model $\Theta_{\bar{v}}$ in the complementary view v.
    E-Step: Learn parameters Θ_v in current view from labeled data D_l and unlabeled data D_u with class probabilities $p_v(y \mid x, \Theta_{\bar{v}})$.

Both algorithms return confidence weighted hypothesis $\frac{1}{2}(f_{(\Theta,1)} + f_{(\Theta,2)})$.

f_2 make an error for a given instance. The *correlation coefficient* of these random variables is defined as

$$\Phi^2 = \frac{\sum_{i=0}^{1}\sum_{j=0}^{1}(P(E_1=i,E_2=j)-P(E_1=i)P(E_2=j))^2}{P(E_1=i)P(E_2=j)}$$

which quantifies whether these events occur independently — in this case, $\Phi^2 = 0$ — or are dependent. In the most extreme case, when the two hypotheses always err at the same time, then $\Phi^2 = 1$. In experiments with gene function prediction and text classification problems, Krogel & Scheffer (2004) have found a clearly negative relationship between the benefit of co-training for a given problem and the error correlation coefficient Φ^2 of the initial classifiers. When the initial classifiers are correlated, for example,with $\Phi^2 \geq 0.3$, then co-training will often deteriorate the classification result instead of improving it.

FUTURE TRENDS

In knowledge discovery and machine learning, one is often interested in discriminative learning. Generative models allow you to easily incorporate unlabeled data into the learning process via the EM algorithm, but model-based learners optimize the wrong utility criterion when the goal is really discriminative learning (for example, see Cozman et al., 2003). Graph-based approaches (Blum & Chawla, 2001) allow the utilization of unlabeled data for discriminative learning, under the mild model assumption that instances with identical classes are more likely to be neighbors than instances with distinct classes. This idea is currently being investigated for a range of applications (Joachims, 2003; Zhu et al., 2003; Blum, Lafferty, Reddy, & Rwebangira, 2004).

The principle of minimizing the disagreement of independent hypothesis is a simple yet powerful mechanism that allows minimization of an upper bound on the error by using only unlabeled data. Exploiting this principle for additional learning tasks such as clustering (Bickel & Scheffer, 2004; Kailing, Kriegel, Pryakhin, & Schubert, 2004), and by more effective algorithms, is a principal challenge that will lead to more powerful and broadly applicable semi-supervised learning algorithms. Algorithms that automatically analyze attribute interactions (Jakulin & Bratko, 2004) will possibly extend the scope of multi-view learning applicable to learning problems for which independent attribute sets are not available a priori.

CONCLUSION

The Expectation Maximization algorithm provides a framework for incorporating unlabeled data into model-based learning. However, the model that maximizes the joint likelihood of labeled and unlabeled data can, in principle, be a worse discriminator than a model that was trained only on labeled data. Mincut algorithms allow the utilization of unlabeled data in discriminative learning; similar to the transductive SVM, only mild assumptions on $p(x \mid y)$ are made.

The multi-view framework provides a simple yet powerful mechanism for utilizing unlabeled data: The disagreement of two independent hypotheses upper bounds the error — it can be minimized with only unlabeled data. A prerequisite of multi-view learning is two independent views; the dependence of views can be quantified and measured by the error correlation coefficient. A small correlation coefficient corresponds to a great expected benefit of multi-view learning. The co-EM algorithm is the most effective multi-view algorithm when the views are independent; co-training is more robust against violations of this independence. Only when dependencies are strong is multi-view learning detrimental.

ACKNOWLEDGMENT

The author is supported by Grant SCHE540/10-1 of the German Science Foundation DFG.

REFERENCES

Abney, S. (2002). Bootstrapping. *Proceedings of the Annual Meeting of the Association for Computational Linguistics.*

Bickel, S., & Scheffer, T. (2004). Multi-view clustering. *Proceedings of the IEEE International Conference on Data Mining.*

Blum, A., & Chawla, S. (2001). Learning from labeled and unlabeled data using graph mincuts. *Proceedings of the International Conference on Machine Learning.*

Blum, A., Lafferty, J., Reddy, R., & Rwebangira, M. (2004). Semi-supervised learning using randomized mincuts. *Proceedings of the International Conference on Machine Learning*.

Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. *Proceedings of the Conference on Computational Learning Theory*.

Brefeld, U., & Scheffer, T. (2004). Co-EM support vector learning. *Proceedings of the International Conference on Machine Learning*.

Collins, M., & Singer, Y. (1999). Unsupervised models for named entity classification. *Proceedings of the Conference on Empirical Methods for Natural Language Processing*.

Cooper, D., & Freeman, J. (1970). On the asymptotic improvement in the outcome of supervised learning provided by additional nonsupervised learning. *IEEE Transactions on Computers, C-19*, 1055-1063.

Cozman, F., Cohen, I., & Cirelo, M. (2003). Semi-supervised learning of mixture models. *Proceedings of the International Conference on Machine Learning*.

Dasgupta, S., Littman, M., & McAllester, D. (2001). PAC generalization bounds for co-training. In T. G. Dietterich, S. Becker, & Z. Ghahramani (Eds.), *Advances in neural information processing systems 14*. Cambridge, MA: MIT Press.

de Sa (1994). *Learning classification with unlabeled data.* Advances of Neural Information Processing Systems.

Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B, 39*.

Jakulin, A., & Bratko, I. (2004). Testing the significance of attribute interactions. *Proceedings of the International Conference on Machine Learning*.

Joachims, T. (1999). Transductive inference for text classification using support vector machines. *Proceedings of the International Conference on Machine Learning*.

Joachims, T. (2003). Transductive learning via spectral graph partitioning. *Proceedings of the International Conference on Machine Learning*.

Kailing, K., Kriegel, H, Pryakhin, A., & Schubert, M. (2004). Clustering multi-represented objects with noise. *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*.

Krogel, M.-A., & Scheffer, T. (2004). Multirelational learning, text mining, and semi-supervised learning for functional genomics. *Machine Learning, 57*(1/2), 61-81.

Miller, D., & Uyar, H. (1997). A mixture of experts classier with learning based on both labelled and unlabelled data. In xxx (Eds.), *Advances in neural information processing systems 9*. Cambridge, MA: MIT Press.

Muslea, I., Kloblock, C., & Minton, S. (2002). Active + semi-supervised learning = robust multi-view learning. *Proceedings of the International Conference on Machine Learning*.

Nigam, K., & Ghani, R. (2000). Analyzing the effectiveness and applicability of co-training. *Proceedings of the International Conference on Information and Knowledge Management*.

Nigam, K., McCallum, A., Thrun, S., & Mitchell, T. (2000). Text classification from labeled and unlabeled documents using EM. *Machine Learning, 39*(2/3).

Seeger, M. (2001). *Learning with labeled and unlabeled data.* Technical report, University of Edinburgh.

Titterington, D., Smith, A., & Makov, U. (1985). *Statistical analysis of finite mixture distributions.* Wiley.

Vapnik, V. (1998). *Statistical learning theory.* Wiley.

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Zhang, T., & Oles, F. (2000). A probability analysis on the value of unlabeled data for classification problems. *Proceedings of the International Conference on Machine Learning*.

Zhu, X., Gharamani, Z., & Lafferty, J. (2003). Semi-supervised learning using Gaussian fields and harmonic functions. *Proceedings of the International Conference on Machine Learning*.

KEY TERMS

Compatibility: Views V_1 and V_2 are compatible if there exist functions f_1 and f_2 such that for a11 x, $f_1(x_1) = f_2(x_2) = f(x)$ where f is the true target function.

Independence: Views V_1 and V_2 are conditionally independent given the class if for all $x = (x_1,x_2)$, $P(x_1,x_2 | y) = P(x_1 | y)\, P(x_2 | y)$.

Labeled Data: A sequence of training instances with corresponding class labels, where the class label is the value to be predicted by the hypothesis.

Multi-View Learning: A family of semi-supervised or unsupervised learning algorithms that can be applied when instances are represented by two sets of features, provided that these sets are conditionally independent given the class and that either set suffices to learn the target concept. By minimizing the disagreement between two independent classifiers, multi-view algorithms minimize an upper bound on the error rate that can be determined without reference to labeled data.

Semi-Supervised Classification: The task of learning a mapping from instances to one of finitely many class labels, coming from labeled data consisting of a sequence of instance-class pairs and unlabeled data consisting of just a sequence of instances.

Supervised Learning: The task of learning a mapping from instances to function values (possibly class labels) from a sequence of pairs of instances and function values.

Unlabeled Data: A sequence of training instances without corresponding class labels.

Unsupervised Learning: The task of learning a model that describes a given data set where the attribute of interest is not available in the data. Often, the model is a mixture model, and the mixture component, from which each instance has been drawn, is not visible in the data.

View: In multi-view learning, the available attributes are partitioned into two disjoint subsets, or views, which are required to be *independent* and *compatible.*

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 1022-1027, copyright 2005 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

Sentiment Analysis of Product Reviews

Cane W. K. Leung
The Hong Kong Polytechnic University, Hong Kong SAR

Stephen C. F. Chan
The Hong Kong Polytechnic University, Hong Kong SAR

INTRODUCTION

Sentiment analysis is a kind of text classification that classifies texts based on the *sentimental orientation* (SO) of opinions they contain. Sentiment analysis of product reviews has recently become very popular in text mining and computational linguistics research. The following example provides an overall idea of the challenge. The sentences below are extracted from a movie review on the Internet Movie Database:

"It is quite boring..... the acting is brilliant, especially Massimo Troisi."

In the example, the author stated that "it" (the movie) is quite boring but the acting is brilliant. Understanding such sentiments involves several tasks. Firstly, evaluative terms expressing opinions must be extracted from the review. Secondly, the SO, or the polarity, of the opinions must be determined. For instance, "boring" and "brilliant" respectively carry a negative and a positive opinion. Thirdly, the opinion strength, or the intensity, of an opinion should also be determined. For instance, both "brilliant" and "good" indicate positive opinions, but "brilliant" obviously implies a stronger preference. Finally, the review is classified with respect to sentiment classes, such as *Positive* and *Negative*, based on the SO of the opinions it contains.

BACKGROUND

Sentiment analysis is also known as opinion mining, opinion extraction and affects analysis in the literature. Further, the terms *sentiment analysis* and *sentiment classification* have sometimes been used interchangeably. It is useful, however, to distinguish between two subtly different concepts. In this article, hence, sentiment analysis is defined as a complete process of extracting and understanding the sentiments being expressed in text documents, whereas sentiment classification is the task of assigning class labels to the documents, or segments of the documents, to indicate their SO.

Sentiment analysis can be conducted at various levels. Word level analysis determines the SO of an opinion word or a phrase (Kamps et al., 2004; Kim and Hovy, 2004; Takamura and Inui, 2007). Sentence level and document level analyses determine the dominant or overall SO of a sentence and a document respectively (Hu and Liu, 2004a; Leung et al., 2008). The main essence of such analyses is that a sentence or a document may contain a mixture of positive and negative opinions. Some existing work involves analysis at different levels. Specifically, the SO of opinion words or phrases can be aggregated to determine the overall SO of a sentence (Hu and Liu, 2004a) or that of a review (Turney, 2002; Dave et al., 2003; Leung et al., 2008).

Most existing sentiment analysis algorithms were designed for binary classification, meaning that they assign opinions or reviews to bipolar classes such as *Positive* or *Negative* (Turney, 2002; Pang et al., 2002; Dave et al., 2003). Some recently proposed algorithms extend binary sentiment classification to classify reviews with respect to multi-point rating scales, a problem known as *rating inference* (Pang and Lee, 2005; Goldberg and Zhu, 2006; Leung et al., 2008). Rating inference can be viewed as a multi-category classification problem, in which the class labels are scalar ratings such as 1 to 5 "stars".

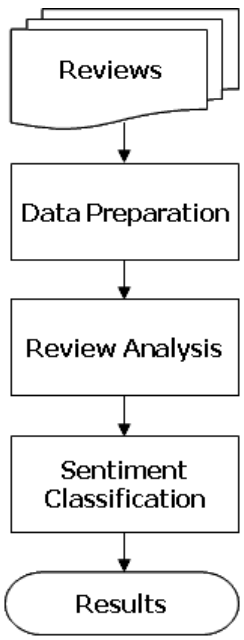
Some sentiment analysis algorithms aim at summarizing the opinions expressed in reviews towards a given product or its features (Hu and Liu, 2004a; Gamon et al., 2005). Note that such *sentiment summarization* also involves the classification of opinions according to their SO as a subtask, and that it is different from classical document summarization, which is about identifying the key sentences in a document to summarize its major ideas.

Sentiment analysis is closely related to *subjectivity analysis* (Wiebe et al., 2001; Esuli and Sebastiani, 2005). Subjectivity analysis determines whether a given text is subjective or objective in nature. It has been addressed using two methods in sentiment analysis algorithms. The first method considers subjectivity analysis a binary classification problem, for example, using *Subjective* and *Objective* as class labels. Pang and Lee (2005) adopted this method to identify subjective sentences in movie reviews. The second method makes use of part-of-speech (POS) information about words to identify opinions (Turney, 2002; Hu and Liu, 2004a; Leung et al., 2008) as previous work on subjectivity analysis suggests that adjectives usually have strong and significant correlation with subjectivity (Bruce and Wiebe, 1999; Wiebe et al., 2001).

MAIN FOCUS

Figure 1 depicts a typical sentiment analysis model. The model takes a collection of reviews as input and processes them using three core steps, *Data Preparation*, *Review Analysis* and *Sentiment Classification*. The results produced by such a model are the classifications of the reviews, the evaluative sentences, or opinions expressed in the reviews.

Figure 1. A typical sentiment analysis model



Data Preparation

The data preparation step performs necessary data preprocessing and cleaning on the dataset for the subsequent analysis. Some commonly used preprocessing steps include removing non-textual contents, markup tags (for HTML pages) and other information that is not required for sentiment analysis, such as review dates and reviewers' names.

Data preparation may also involve the sampling of reviews for building a classifier. Positive reviews often predominate in review datasets as reported in several studies (e.g. Turney, 2002; Dave et al., 2003; Gamon et al., 2005). Some researchers therefore used review datasets with balanced class distributions when training classifiers to help demonstrate the performance of their algorithms (Pang et al., 2002; Leung et al., 2008).

Review Analysis

The review analysis step analyzes the linguistic features of reviews so that interesting information, including opinions and/or product features, can be identified. This step often applies various computational linguistics tasks to reviews first, and then extracts opinions and product features from the processed reviews. Two commonly adopted tasks for review analysis are POS tagging and negation tagging. POS tagging helps identifying interesting words or phrases having particular POS tags or patterns from reviews (Turney, 2002; Hu and Liu, 2004a; Leung et al., 2008), while negation tagging is used to address the contextual effect of negation words, such as "not", in a sentence (Pang et al., 2002; Dave et al., 2003; Leung et al., 2008). For example, "good" and "not good" obviously indicate opposite SO. Given the term "not good", negation tagging recognizes the existence of the word "not" and adds a special negation tag to the word "good" based on some heuristics.

The review analysis step then proceeds to extract opinions and/or product features from the processed reviews. The opinions or features extracted may be *n*-grams, which are *n* adjacent or nearby words in a sentence (e.g. Turney, 2002). Pang et al. (2002) make use of corpus statistics and human introspection to decide

terms that may appear in reviews. Various algorithms adopt a more common method that extracts words or phrases having particular POS tags or patterns as opinions and product features as noted (Turney, 2002; Dave et al., 2003; Takamura and Inui, 2007, Leung et al., 2008).

While Hu and Liu (2004b) also make use of POS tags, they adapted the idea of frequent itemsets discovery in association rule mining to product feature extraction. In the context of their work, an itemset is a set of words that occurs together, and a "transaction" contains nouns or noun phrases extracted from a sentence of a review. They used the CBA association rule miner (Liu et al., 1998) to mine frequent itemsets, and considered each resulting itemset to be a product feature. They then processed a review sentence by sentence. If a sentence contains a frequent feature, they extracted its nearby adjective as an opinion. They also proposed methods for pruning redundant features and for identifying infrequent features.

Sentiment Classification

There are two major approaches to sentiment classification, known as the *SO approach* and the *machine learning approach*. The following subsections describe the overall idea and representative techniques of each of the approaches.

SO Approach

The SO approach involves two subtasks. The first subtask is to determine the SO of the opinions extracted from reviews in the Review Analysis step, while the second subtask is to determine the overall SO of a sentence or a review based on the SO of the opinions it contains.

Turney (2002) proposed an unsupervised SO determination method that computes the SO of an opinion phrase as the Pointwise Mutual Information (PMI) between the phrase and two *seed adjectives*, "excellent" and "poor". Such information is collected using a generic search engine. Specifically, the phrase is likely to represent a positive (resp. negative) sentiment if it co-occurs frequently with the word "excellent" (resp. "poor"). The average of the SO of all opinion phrases in a review is computed, and the review is classified as *Positive* if the average is positive; and *Negative* otherwise.

Hu and Liu (2004a) presented a word-similarity-based algorithm that utilizes the semantical relationship between words to predict SO. Their bootstrapping algorithm depends on a small set of seed adjectives having known SO, such as "great" for *Positive* and "bad" for *Negative*, and automatically expands the set using the synonym and antonym sets in WordNet (Miller et al., 1990), assuming that *semantical similarity implies sentimental similarity*. Specifically, the SO of synonyms is assumed to be the same, whereas that of antonyms is opposite to each other. After predicting the SO of opinions, their algorithm classifies a sentence based on the dominant SO of the opinions in the sentence. Kamps et al. (2004) and Kim and Hovy (2004) also described word-similarity-based methods for determining SO, but their studies deal with word-level classification.

Leung et al. (2006b) suggest that semantical similarity may not imply sentimental similarity in sentiment analysis, based on statistical observations from a movie review corpus. They therefore proposed a relative-frequency-based method for determining the SO of an opinion. Their method estimates the SO and opinion strength of a word with respect to a sentiment class as its relative frequency of appearance in that class. For example, if the word "best" appeared 8 times in positive reviews and 2 times in negative reviews, its opinion strength with respect to *Positive* SO is then 8/(8+2) = 0.8. Leung et al. (2008) deals with the rating inference problem, which classifies reviews with respect to rating scales. They hypothesized that some product features may be more important for determining the rating of a review, and therefore assign weights to opinions according to the estimated importance of their associated product features. They compute the weighted average SO of opinions in a review, and then rate the review by mapping the weighted average onto an *n*-point rating scale.

Machine Learning Approach

The machine learning approach is similar to topic classification, with the topics being sentiment classes such as *Positive* and *Negative* (Pang et al., 2002). It works by breaking down a review into words or phrases, representing the review as a document vector (bag-of-words model), and then classifying the reviews based on the document vectors.

Pang et al. (2002) investigated whether binary sentiment classification can be addressed using standard topic classification techniques. They applied three classifiers, including Naïve Bayes, Support Vector Machines (SVM) and Maximum Entropy, to a movie review corpus. They also attempted to incorporate various features of the reviews into the standard bag-of-words model, such as the positions of words in the reviews, but the performance of the three classifiers was found inferior to those reported for topic classification. Pang and Lee concluded that sentiment classification is more difficult than topic classification, and that discourse analysis of reviews is necessary for more accurate sentiment analysis.

Pang and Lee (2005) formulated rating inference as a metric-labeling problem. They first applied two *n*-ary classifiers, including one-vs-all (OVA) SVM and SVM regression, to classify reviews with respect to multi-point rating scales. They then used a metric-labeling algorithm to explicitly alter the results of the *n*-ary classifiers to ensure that similar items receive similar labels, determined using a similarity function. While term overlapping is a commonly-used similarity function in topic classification, it does not seem effective in identifying reviews having similar ratings (Pang and Lee, 2005). They therefore proposed the Positive-Sentence Percentage (PSP) similarity function, computed as the number of positive sentences divided by the number of subjective sentences in a review. Experimental results in general show that using metric-labeling with PSP improves the performance of the *n*-ary classifiers. Goldberg and Zhu (2006) later extended Pang and Lee's work using transductive semi-supervised learning. They demonstrated that unlabeled reviews (those without user-specified ratings) can help improve classification accuracy.

Zhu and Goldberg (2007) proposed a kernel regression algorithm utilizing *order preferences* of unlabeled data, and successfully applied the algorithm to sentiment classification. The order preference of a pair of unlabeled data, x_i and x_j, indicates that x_i is preferred to x_j to some degree, even though the exact preferences for x_i and x_j are unknown. In the context of sentiment analysis, for example, given two reviews, one may be able to determine which review is more positive than the other without knowing the exact ratings associated with the reviews. Zhu and Goldberg applied their algorithm to the rating inference problem, and empirically showed that order preferences improved rating inference performance over standard regression.

FUTURE TRENDS

Most existing algorithms adopt generic opinion and product feature extraction methods, such as methods based on POS tags, without considering the properties of the domain items concerned. While such generic methods allow easy adaptation of a sentiment analysis algorithm to various domains, the performance achieved by the same algorithm was often found to vary significantly when being applied to datasets from different domains (e.g. Turney, 2002; Aue and Gamon, 2005). This, currently being addressed as a domain adaptation issue (Blitzer et al., 2007), reveals a need for more intelligent sentiment analysis models that utilize domain knowledge when extracting opinions and product features from reviews.

An emerging trend regarding sentiment classification is the paradigm shift from binary classification (e.g. *Positive* vs. *Negative*) to multi-point rating inference (e.g. 1-5 "stars"). Recent studies suggest that rating inference shall not be tackled as a classical multi-category classification problem because the ordering of class labels in rating inference is essential (Okanohara and Tsujii, 2005; Pang and Lee, 2005; Zhu and Goldberg, 2007). This opens up an interesting direction in ordered multi-category classification for future research.

Another interesting research direction is related to the utilization of the sentiments learnt from product reviews. Sentiment analysis has been used to support business and customer decision making by assisting users to explore customer opinions on products that they are interested in (Yamanishi and Li, 2002; Hu and Liu, 2004a). Leung et al. (2008) recently discussed the potential use of sentiment analysis to augment ratings for collaborative filtering (CF), which is also a popular research topic in and application of data mining.

CF provides personalized recommendations to a user based on his/her preferences and the preferences of other users having similar tastes (Leung et al., 2006a). A CF-based system operates on a database of user preferences collected either explicitly by asking users to give scalar ratings on items that they have examined, or implicitly by capturing users' interactions with the system (e.g. purchase histories). Using sentiment analysis to augment ratings for CF on the one hand allows

CF to use product reviews as an additional source of user preferences. On the other hand, it enables existing review hubs to utilize the user preferences learnt from reviews for personalization purpose. In view of these advantages, integrating sentiment analysis and CF is expected to be of high interests to data and text mining practitioners.

CONCLUSION

Sentiment analysis deals with the classification of texts based on the sentiments they contain. This article focuses on a typical sentiment analysis model consisting of three core steps, namely data preparation, review analysis and sentiment classification, and describes representative techniques involved in those steps.

Sentiment analysis is an emerging research area in text mining and computational linguistics, and has attracted considerable research attention in the past few years. Future research shall explore sophisticated methods for opinion and product feature extraction, as well as new classification models that can address the ordered labels property in rating inference. Applications that utilize results from both sentiment analysis and CF are also expected to emerge in the near future.

REFERENCES

Aue, A., & Gamon, M. (2005). Customizing sentiment classifiers to new domains: A case study. *Proceedings of Recent Advances in Natural Language Processing*.

Blitzer, J., Dredze, M., & Pereira, F. (2007) Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. *Proceedings of the 45th Annual Meeting of the ACL*. Retrieved June 23, 2007, from http://acl.ldc.upenn.edu/P/P07/P07-1056.pdf

Bruce, R., & Wiebe, J. (1999). Recognizing subjectivity: A case study of manual tagging. *Natural Language Engineering*, 5(2), 187-205.

Dave, K., Lawrence, S., & Pennock, D. M. (2003). Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. *Proceedings of 12th International World Wide Web Conference*.

Esuli, A., & Sebastiani, F. (2005). Determining the semantic orientation of terms through gloss classification. *Proceedings of the ACM Conference on Information and Knowledge Management*, pp. 617-624.

Gamon, M., Aue, A., Corston-Oliver, S., & Ringger, E. (2005). Pulse: Mining customer opinions from free text. *Lecture Notes in Computer Science*, vol. 3646, pp. *121-132*.

Goldberg, A. B., & Zhu, X. (2006). Seeing stars when there aren't many stars: Graph-based semi-supervised learning for sentiment categorization. *Proceedings of TextGraphs Workshop,* pp. 45-52.

Hu, M., & Liu, B. (2004a). Mining and summarizing customer reviews. *Proceedings of 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 168-177.

Hu, M., & Liu, B. (2004b). Mining opinion features in customer reviews. *Proceedings of 19th National Conference on Artificial Intelligence*, pp. 755-760.

Kamps, J., Marx, M., Mokken, R. J., & de Rijke, M. (2004). Using WordNet to measure semantic orientation of adjectives. *Proceedings of 4th International Conference on Language Resources and Evaluation*, vol. VI, 1115-1118.

Kim, S.-M., & Hovy, E. H. (2004). Determining the sentiment of opinions. *Proceedings of 20th International Conference on Computational Linguistics*, pp. 1367-1373.

Leung, C. W. K., Chan, S. C. F., & Chung, F. L. (2006a). A Collaborative Filtering Framework Based on Fuzzy Association Rules and Multiple-Level Similarity. *Knowledge and Information Systems (KAIS)*, 10(3), 357-381.

Leung, C. W. K., Chan, S. C. F., & Chung, F. L. (2006b). Integrating collaborative filtering and sentiment analysis: A rating inference approach. *Proceedings of ECAI 2006 Workshop on Recommender Systems*, pp. 62-66.

Leung, C. W. K., Chan, S. C. F., & Chung, F. L. (2008). Evaluation of a rating inference approach to utilizing textual reviews for collaborative recommendation. *Cooperative Internet Computing*, World Scientific. Retrieved May 23, 2007, from http://www4.comp.polyu.edu.hk/~cswkleung/pub/cic_evalRatingInference.pdf

Liu, B., Hsu, W., & Ma, Y. (1998). Integrating classification and association rule mining. *Proceedings of Knowledge Discovery and Data Mining*, pp. 80-86.

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., & Miller, K. J. (1990). Introduction to WordNet: An on-line lexical database. *International Journal of Lexicography (Special Issue)*, 3(4), 235-244.

Okanohara, D., & Tsujii, J. (2005). Assigning polarity scores to reviews using machine learning techniques. In R. Dale, K.-F. Wong, J. Su and O. Y. Kwong (Eds.), *Natural Language Processing - IJCNLP 2005*, Springer-Verlag, pp. 314-325.

Pang, B., & Lee, L. (2005) Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. *Proceedings of 43rd Annual Meeting of the ACL*, pp. 115-124.

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pp. 79-86.

Takamura, H., Inui, T., & Okumura, M. (2007) Extracting semantic orientations of phrases from dictionary. *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the ACL*, pp. 292-299.

Turney, P (2002). Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. *Proceedings of 40th Annual Meeting of the ACL*, pp. 417-424.

Wiebe, J., Bruce, R., Bell, M., Martin, M., & Wilson, T. (2001). A corpus study of evaluative and speculative language. *Proceedings of 2nd ACL SIGdial Workshop on Discourse and Dialogue*. Aalborg, Denmark.

Yamanishi, K., & Li, H. (2002). Mining open answers in questionnaire data. *IEEE Intelligent Systems*, 17(5), 58-63.

Zhu, X., & Goldberg, A. B. (2007). Kernel Regression with Order Preferences. *Proceedings of the Twenty-Second Conference on Artificial Intelligence (AAAI-07)*, Retrieved May 23, 2007, from http://www.cs.wisc.edu/~jerryzhu/pub/orderssl_aaai07.pdf

KEY TERMS

Collaborative Filtering: A recommendation technique that provides personalized recommendations to a user based on his/her expressed interests and the interests of other users having similar preferences.

Opinion Strength: Indicates how strong the opinion is given its sentimental orientation. It is also known as the intensity of an opinion.

Rating Inference: Refers to sentiment classification with respect to multi-point rating scales. It can be viewed as an *n*-ary classification problem in which the class labels are scalar ratings.

Sentiment Analysis: The process of analyzing the sentiments expressed in texts, and then classifying and/or summarizing the sentiments based on their polarity.

Sentiment Classification: A core step in sentiment analysis that classifies a text with respect to sentiment classes, such as *Positive* and *Negative*. An *n*-ary sentiment classification problem is also known as rating inference.

Sentiment Summarization: Summarizes the opinions expressed in a document or in a set of documents towards a product.

Sentimental Orientation: Indicates the polarity, such as *Positive* or *Negative*, of a text known to be subjective.

Subjectivity Analysis: Determines whether a text is subjective or objective (factual) in nature.

Sequential Pattern Mining

Florent Masseglia
INRIA Sophia Antipolis, France

Maguelonne Teisseire
University of Montpellier II, France

Pascal Poncelet
Ecole des Mines d'Alès, France

INTRODUCTION

Sequential pattern mining deals with data represented as sequences (a sequence contains sorted sets of items). Compared to the association rule problem, a study of such data provides "inter-transaction" analysis (Agrawal & Srikant, 1995). Applications for sequential pattern extraction are numerous and the problem definition has been slightly modified in different ways. Associated to elegant solutions, these problems can match with real-life timestamped data (when association rules fail) and provide useful results.

BACKGROUND

In (Agrawal & Srikant, 1995) the authors assume that we are given a database of customer's transactions, each of which having the following characteristics: sequence-id or customer-id, transaction-time and the item involved in the transaction. Such a database is called a base of data sequences. More precisely, each transaction is a set of items (itemset) and each sequence is a list of transactions ordered by transaction time. For efficiently aiding decision-making, the aim is to obtain typical behaviors according to the user's viewpoint. Performing such a task requires providing data sequences in the database with a support value giving its number of actual occurrences in the database. A frequent sequential pattern is a sequence whose statistical significance in the database is above user-specified threshold. Finding all the frequent patterns from huge data sets is a very time-consuming task. In the general case, the examination of all possible combination is intractable and new algorithms are required to focus on those sequences that are considered important to an organization.

Sequential pattern mining is applicable in a wide range of applications since many types of data are in a time-related format. For example, from a customer purchase database a sequential pattern can be used to develop marketing and product strategies. By way of a Web Log analysis, data patterns are very useful to better structure a company's website for providing easier access to the most popular links (Kosala & Blockeel, 2000). We also can notice telecommunication network alarm databases, intrusion detection (Hu & Panda, 2004), DNA sequences (Zaki, 2003), etc.

MAIN THRUST

Definitions related to the sequential pattern extraction will first be given. They will help understanding the various problems and methods presented hereafter.

Definitions

The item is the basic value for numerous data mining problems. It can be considered as the object bought by a customer, or the page requested by the user of a website, etc. An itemset is the set of items that are grouped by timestamp (e.g. all the pages requested by the user on June 04, 2004). A data sequence is a sequence of itemsets associated to a customer. In table 1, the data sequence of C2 is the following: "(Camcorder, MiniDV) (DVD Rec, DVD-R) (Video Soft)" which means that the customer bought a *camcorder* and *miniDV* the same day, followed by a *DVD recorder* and *DVD-R* the day after, and finally a *video software* a few days later.

A sequential pattern is included in a data sequence (for instance "(MiniDV) (Video Soft)" is included in the data sequence of C2, whereas "(DVD Rec) (Camcorder)" is not included according to the order of the

Table 1. Data sequences of four customers over four days

Cust	June 04, 2004	June 05, 2004	June 06, 2004	June 07, 2004
C1	Camcorder, MiniDV	Digital Camera	MemCard	USB Key
C2	Camcorder, MiniDV	DVD Rec, DVD-R		Video Soft
C3	DVD Rec, DVD-R	MemCard	Video Soft	USB Key
C4		Camcorder, MiniDV	Laptop	DVD Rec, DVD-R

timestamps). The minimum support is specified by the user and stands for the minimum number of occurrences of a sequential pattern to be considered as frequent. A maximal frequent sequential pattern is included in at least "minimum support" data sequences and is not included in any other frequent sequential pattern. Table 1 gives a simple example of 4 customers and their activity over 4 days in a shop. With a minimum support of "50%" a sequential pattern can be considered as frequent if it occurs at least in the data sequences of 2 customers (2/4). In this case a maximal sequential pattern mining process will find three patterns:

- **S1:** "(Camcorder, MiniDV) (DVD Rec, DVD-R)"
- **S2:** "(DVD Rec, DVD-R) (Video Soft)"
- **S3:** "(Memory Card) (USB Key)"

One can observe that S1 is included in the data sequences of C2 and C4, S2 is included in those of C2 and C3, and S3 in those of C1 and C2. Furthermore the sequences do not have the same length (S1 has length 4, S2 has length 3 and S3 has length 2).

Methods for Mining Sequential Patterns

The problem of mining sequential patterns is stated in (Agrawal & Srikant, 1995) and improved, both for the problem and the method, in (Srikant & Agrawal, 1996). In the latter, the GSP algorithm is based on a breadth-first principle since it is an extension of the A-priori model to the sequential aspect of the data. GSP uses the "Generating-Pruning" method defined in (Agrawal, Imielinski, & Swami, 1993) and performs in the following way. A candidate sequence of length (k+1) is generated from two frequent sequences, s_1 and s_2, having length k, if the subsequence obtained by pruning the first item of s_1 is the same as the subsequence obtained by pruning the last item of s_2. With the example in Table 1, and k=2, let s_1 be "(DVD Rec, DVD-R)" and s_2 be "(DVD-R) (Video Soft)", then the candidate sequence will be "(DVD Rec, DVD-R) (Video Soft)" since the subsequence described above (common to s_1 and s_2) is "(DVD-R)". Another method based on the Generating-Pruning principle is PSP (Massegli, Cathala, & Poncelet, 1998). The main difference to GSP is that the candidates as well as the frequent sequences are managed in a more efficient structure. The methods presented so far are designed to depend as little as possible on main memory. The methods presented thereafter need to load the database (or a rewriting of the database) in main memory. This results in efficient methods when the database can fit into the memory.

In (Zaki, 2001), the authors proposed the SPADE algorithm. The main idea in this method is a clustering of the frequent sequences based on their common prefixes and the enumeration of the candidate sequences, thanks to a rewriting of the database (loaded in main memory). SPADE needs only three database scans in order to extract the sequential patterns. The first scan aims at finding the frequent items, the second at finding the frequent sequences of length 2 and the last one associate to frequent sequences of length 2, a table of the corresponding sequences id and itemsets id in the database (e.g. data sequences containing the frequent sequence and the corresponding timestamp). Based on this representation in main memory, the support of the candidate sequences of length k is the result of join operations on the tables related to the frequent sequences of length (k-1) able to generate this candidate (so, every operation after the discovery of frequent sequences having length 2 is done in memory). SPAM (Ayres, Flannick, Gehrke, & Yiu, 2002) is another method which needs to represent the database in the main memory. The authors proposed a vertical bitmap representation of the database for both candidate representation and support counting.

An original approach for mining sequential patterns aims at recursively projecting the data sequences into smaller databases. Proposed in (Han, et al., 2000), FreeSpan is the first algorithm considering the pattern-projection method for mining sequential patterns. This work has been continued with PrefixSpan, (Pei, et al., 2001), based on a study about the number of candidates proposed by a Generating-Pruning method. Starting from the frequent items of the database, PrefixSpan generates projected databases with the remaining data-sequences. The projected databases thus contain suffixes of the data-sequences from the original database, grouped by prefixes. The process is recursively repeated until no frequent item is found in the projected database. At this level the frequent sequential pattern is the path of frequent items driving to this projected database.

Closed Sequential Patterns

A closed sequential pattern is a sequential pattern included in no other sequential pattern having exactly the same support. Let us consider the database illustrated in Table 1. The frequent sequential pattern "(DVD Rec) (Video Soft)" is not closed because it is included in the sequential pattern S2 which has the same support (50%). On the other hand, the sequential pattern "(Camcorder, MiniDV)" (with a support of 75%) is closed because it is included in other sequential patterns but with a different support (for instance, S1, which has a support of 50%). The first algorithm designed to extract closed sequential patterns is CloSpan (Yan, Han, & Afshar, 2003) with a detection of non-closed sequential patterns avoiding a large number of recursive calls. CloSpan is based on the detection of frequent sequences of length 2 such that "*A* always occurs before/after *B*". Let us consider the database given in Table 1. We know that "(DVD Rec) (Video Soft)" is a frequent pattern. The authors of CloSpan proposed relevant techniques to show that "(DVD-R)" always occurs before "(Video Soft)". Based on this observation CloSpan is able to find that "(DVD Rec, DVD-R) (Video Soft)" is frequent without anymore scans over the database. BIDE (Wang & Han, 2004) extends the previous algorithm in the following way. First, it adopts a novel sequence extension, called BI-Directional Extension, which is used both to grow the prefix pattern and to check the closure property. Second, in order to prune the search space more deeply than previous approaches, it proposes a BackScan pruning method. The main idea of this method is to avoid extending a sequence by detecting in advance that the extension is already included in a sequence.

Incremental Mining of Sequential Patterns

As databases evolve, the problem of maintaining sequential patterns over a significantly long period of time becomes essential since a large number of new records may be added to a database. To reflect the current state of the database, in which previous sequential patterns would become irrelevant and new sequential patterns might appear, new efficient approaches were proposed. (Masseglia, Poncelet, & Teisseire, 2003) proposes an efficient algorithm, called ISE, for computing the frequent sequences in the updated database. ISE minimizes computational costs by re-using the minimal information from the old frequent sequences, i.e. the support of frequent sequences. The main new feature of ISE is that the set of candidate sequences to be tested is substantially reduced. The SPADE algorithm was extended into the ISM algorithm (Parthasarathy, Zaki, Ogihara, & Dwarkadas., 1999). In order to update the supports and enumerate frequent sequences, ISM maintains "maximally frequent sequences" and "minimally infrequent sequences" (also known as negative border). KISP (Lin and Lee, 2003) also proposes to take advantage of the knowledge previously computed and generates a knowledge base for further queries about sequential patterns of various support values.

Extended Problems Based on the Sequential Pattern Extraction

Motivated by the potential applications for the sequential patterns, numerous extensions of the initial definition have been proposed which may be related to the addition of constraints or to the form of the patterns themselves. In (Pei, Han, & Wang, 2002) the authors enumerate some of the most useful constraints for extracting sequential patterns. These constraints can be considered as filters applied to the extracted patterns, but most methods generally take them into account during the mining process. These filters may concern the items ("extract patterns containing the item *Camcorder* only") or the length of the pattern, regular expressions describing the pattern, and so on.

The definition of the sequential patterns has also been adapted by some research work. For instance (Kum, Pei, Wang, & Duncan, 2003) proposed ApproxMap to mine approximate sequential patterns. ApproxMap first proposes to cluster the data sequences depending on their items. Then for each cluster ApproxMap allows extraction of the approximate sequential patterns related to this cluster. Let us consider the database in Table 1 as a cluster. The first step of the extraction process is to provide the data sequences of the cluster with an alignment similar to those of bioinformatics. Table 2 illustrates such an alignment.

The last sequence in Table 2 represents the weighted sequence obtained by ApproxMap on the sequences of Table 1. With a support of 50%, the weighted sequence gives the following approximate pattern: "(Camcorder: 3, MiniDV: 3) (DVD Rec: 3, DVD-R: 3) (MemCard: 2) (Video Soft: 2) (USB Key: 2)". It is interesting to observe that this sequential pattern does not correspond to any of the recorded behavior, whereas it represents a trend for this kind of customer.

FUTURE TRENDS

Today several methods are available for efficiently discovering sequential patterns according to the initial definition. Such patterns are widely applicable for a large number of applications. Specific methods, widely inspired from previous algorithms, exist in a wide range of domains. Nevertheless, existing methods have to be reconsidered since handled data is much more complex. For example, existing algorithms consider that data is binary and static. Today, according to the huge volume of data available, stream data mining represents an emerging class of data-intensive applications where data flows in and out dynamically. Such applications also need very fast or even real-time responses (Giannella, Han, Pei, Yan, & Yu, 2003; Cai et al., 2004). In order to increase the immediate usefulness of sequential rules, it is very important to consider much more information. Hence, by associating sequential patterns with a customer category or multi-dimensional information, the main objective of multi-dimensional sequential pattern mining is to provide the end-user with more useful classified patterns (Pinto et al., 2001). With such patterns, an auto-dealer would find, for example, an enriched sequential rule stating that "*Customers who bought an SUV on monthly payment installments 2 years ago are likely to respond favorably to a trade-in now*".

CONCLUSION

Since they have been defined in 1995, sequential patterns have received a great deal of attention. First work on this topic focused on improving the efficiency of the algorithms either with new structures, new representations or by managing the database in the main memory. More recently extensions were proposed by taking into account constraints associated with real life applications. In fact, the increasing contributions on sequential pattern mining are mainly due to their adaptability to such applications. The management of timestamp within the recorded data is a difficulty for designing algorithms; on the other hand this is the reason why sequential pattern mining is one of the most promising technologies for the next generation of knowledge discovery problems.

REFERENCES

Agrawal, R., Imielinski, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. *Proceedings of the ACM SIGMOD International*

Table 2. Alignment proposed for the data sequences of Table 1.

Camcorder, MiniDV	DigiCam		MemCard		USB Key
Camcorder, MiniDV		DVD Rec, DVD-R		Video Soft	
		DVD Rec, DVD-R	MemCard	Video Soft	USB Key
Camcorder, MiniDV	Laptop	DVD Rec, DVD-R			
Camcorder: 3 MiniDV: 3	DigiCam: 1 Laptop: 1	DVD Rec: 3 DVD-R: 3	MemCard: 2	Video Soft: 2	USB Key: 2

Conference on Management of Data (pp. 207-216), Washington, D.C, USA.

Agrawal, R., & Srikant, R. (1995). Mining sequential patterns. *Proceeding of the 11th International Conference on Data Engineering* (pp. 3-14), Taipei, Taiwan.

Ayres, J., Flannick, J., Gehrke, J., & Yiu, T. (2002). Sequential pattern mining using bitmap representation. *Proceedings of the 8th International Conference on Knowledge Discovery and Data Mining* (pp. 429-435), Alberta, Canada.

Cai, Y., Clutter, D., Pape, G., Han, J., Welge, M., & Auvil, L. (2004). MAIDS: Mining alarming incidents from data streams. *Proceedings of the ACM SIGMOD International Conference on Management of Data* (pp. 919-920), Paris, France.

Giannella, G., Han, J., Pei, J., Yan, X., & Yu, P. (2003). Mining frequent patterns in data streams at multiple time granularities. In H. Kargupta, A. Joshi, K. Sivakumar & Y. Yesha (Eds.), *Next generation data mining* (chap. 3). MIT Press.

Han, J., Pei, J., Mortazavi-asl, B., Chen, Q., Dayal, U., & Hsu, M. (2000). FreeSpan: Frequent pattern-projected sequential pattern mining. *Proceedings of the 6th International Conference on Knowledge Discovery and Data Mining* (pp. 355-359), Boston, USA.

Hu, Y., & Panda, B. (2004). A Data mining approach for database intrusion detection. *Proceedings of the 19 th ACM Symposium on Applied Computing* (pp. 711-716), Nicosia, Cyprus.

Kosala, R., & Blockeel, H. (2000). Web mining research: A survey. *ACM SIGKDD Explorations, 2*(1), 1-15.

Kum, H.-C., Pei, J., Wang, W., & Duncan, D. (2003). ApproxMAP: Approximate mining of consensus sequential patterns. *Proceedings of the 3rd SIAM International Conference on Data Mining* (pp. 311-315), San Francisco, CA.

Lin, M., & Lee, S. (2003). Improving the efficiency of interactive sequential pattern mining by incremental pattern discovery. *Proceedings of the 36th Annual Hawaii International Conference on System Sciences* (p. 68), Big Island, USA, CDROM.

Masseglia, F., Cathala, F., & Poncelet, P. (1998). The PSP approach for mining sequential patterns. *Proceedings of the 2nd European Symposium on Principles of Data Mining and Knowledge Discovery* (pp. 176-184), Nantes, France.

Masseglia, F., Poncelet, P., & Teisseire, M. (2003). Incremental mining of sequential patterns in large databases. *Data and Knowledge Engineering, 46*(1), 97-121.

Parthasarathy, S., Zaki, M., Ogihara, M., & Dwarkadas, S. (1999). Incremental and interactive sequence mining. *Proceedings of the 8th International Conference on Information and Knowledge Management* (pp. 251-258), Kansas City, USA.

Pei, J., Han, J., Mortazavi-asl, B., Pinto, H., Chen, Q., Dayal, U., et al. (2001). PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth. *Proceedings of 17th International Conference on Data Engineering* (pp. 215-224), Heidelberg, Germany.

Pei, J., Han, J., & Wang, W. (2002). Mining sequential patterns with constraints in large databases. *Proceedings of the11th Conference on Information and Knowledge Management* (pp. 18-25), McLean, USA.

Pinto, H., Han, J., Pei, J., Wang, K., Chen, Q., & Dayal, U. (2001). Multi-dimensional sequential pattern mining. *Proceedings of the 10th International Conference on Information and Knowledge Management* (pp. 81-88), Atlanta, USA.

Srikant, R., & Agrawal, R. (1996). Mining sequential patterns: Generalizations and performance improvements. *Proceeding of the 5th International Conference on Extending Database Technology* (pp. 3-17), Avignon, France.

Wang, J., & Han, J. (2004). BIDE: Efficient mining of frequent closed sequences. *Proceedings of the 20th International Conference of Data Engineering* (pp. 79-90), Boston, USA.

Yan, X., Han, J., & Afshar, R. (2003). CloSpan: Mining closed sequential patterns in large databases. *Proceedings of the 3rd SIAM International Conference on Data Mining*, San Francisco, CA.

Zaki, M. (2001). SPADE: An efficient algorithm for mining frequent sequences. *Machine Learning, 42*(1/2), 31-60.

Zaki, M. (2003). Mining data in bioinformatics. In N. Ye (Ed.), *Handbook of data mining* (pp. 573-596), Lawrence Earlbaum Associates.

KEY TERMS

Apriori: The method of generating candidates before testing them during a scan over the database, insuring that if a candidate may be frequent then it will be generated. See also *Generating-Pruning*.

Breadth-First: The method of growing the intermediate result by adding items both at the beginning and the end of the sequences. See also *Generating-Pruning*

Closed Sequential Pattern: A frequent sequential pattern that is not included in another frequent sequential pattern having exactly the same support.

Data Sequence: The sequence of itemsets representing the behavior of a client over a specific period. The database involved in a sequential pattern mining process is a (usually large) set of data sequences.

Depth-First: The method of generating candidates by adding specific items at the end of the sequences. See also *Generating-Pruning*.

Generating-Pruning: The method of finding frequent sequential patterns by *generating* candidates sequences (from size 2 to the maximal size) step by step. At each step a new generation of candidates having the same length is generated and tested over the databases. Only frequent sequences are kept (*pruning*) and used in the next step to create a new generation of (longer) candidate sequences.

Itemset: Set of items that occur together.

Maximal Frequent Sequential Pattern: A sequential pattern included in at least *n* data sequences (with *n* the minimum support specified by the user). A sequential pattern is maximal when it is not included in another frequent sequential pattern. A frequent sequential pattern may represent, for instance, a frequent behavior of a set of customers, or a frequent navigation of the users of a Web site.

Negative Border: The collection of all sequences that are not frequent but both of whose generating sub sequences are frequent.

Sequential Pattern: A sequence included in a data sequence such that each item in the sequential pattern appears in this data sequence with respect to the order between the itemsets in both sequences.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 523-528, copyright 2005 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

Soft Computing for XML Data Mining

K. G. Srinivasa
M S Ramaiah Institute of Technology, Bangalore, India

K. R. Venugopal
University Visvesvaraya College of Engineering, Bangalore, India

L. M. Patnaik
Indian Institute of Science, Bangalore, India

INTRODUCTION

Efficient tools and algorithms for knowledge discovery in large data sets have been devised during the recent years. These methods exploit the capability of computers to search huge amounts of data in a fast and effective manner. However, the data to be analyzed is imprecise and afflicted with uncertainty. In the case of heterogeneous data sources such as text, audio and video, the data might moreover be ambiguous and partly conflicting. Besides, patterns and relationships of interest are usually vague and approximate. Thus, in order to make the information mining process more robust or say, human-like methods for searching and learning it requires tolerance towards imprecision, uncertainty and exceptions. Thus, they have approximate reasoning capabilities and are capable of handling partial truth. Properties of the aforementioned kind are typical soft computing. *Soft computing* techniques like *Genetic Algorithms* (GA), Artificial Neural Networks, Fuzzy Logic, Rough Sets and *Support Vector Machines* (SVM) when used in combination was found to be effective. Therefore, *soft computing* algorithms are used to accomplish data mining across different applications (Mitra S, Pal S K & Mitra P, 2002; Alex A Freitas, 2002).

Extensible Markup Language (XML) is emerging as a de facto standard for information exchange among various applications of World Wide Web due to XML's inherent data self-describing capacity and flexibility of organizing data. In XML representation, the semantics are associated with the contents of the document by making use of self describing tags which can be defined by the users. Hence XML can be used as a medium for interoperability over the Internet. With these advantages, the amount of data that is being published on the Web in the form of XML is growing enormously and many naïve users find the need to search over large XML document collections (Gang Gou & Rada Chirkova, 2007; Luk R et al., 2000).

BACKGROUND

The SVM is an efficient and principled method used for classification and regression purposes. The SVM is capable of classifying linearly separable and non-linearly separable data. GA is an effective technique for searching enormous, possibly unstructured solution spaces. The human search strategy which is efficient for small documents is not viable when performing search over enormous amounts of data. Hence, making search engines cognizant of the search strategy using GA can help in fast and accurate search over large document collections.

The topic categorization of XML documents poses several new challenges. The tags in XML represent the semantics of the contents of the document and thus are more significant than the contents during the process of classification. Therefore, a general framework which assigns equal priority to both, the tags and the contents of an XML document will not be able to exhibit any significant performance improvement. Thus, a *topic categorization* framework with prominence to tags will be highly efficient. The possibility of topic categorization of XML documents using SVM is explored in (Srinivasa K G et al., 2005).

A *Selective Dissemination* of Information (SDI) system helps users to cope with the large amount of information by automatically disseminating the knowledge to the users in need of it. Therefore, the selective dissemination is the task of dispatching the documents

to the users based on their interests. Such systems maintain user profiles to judge the interests of the users and their information needs. The new documents are filtered against the user profiles, and the relevant information is delivered to the corresponding users. In XML documents, the utilization of user defined tags is of great importance to improve the effectiveness of the dissemination task. The possibility of *selective dissemination* of XML documents based on a user model using Adaptive GAs is addressed in (Srinivasa K G et al., 2007:IOS Press).

The keyword search over XML documents poses many new challenges. First, the result of a search over XML documents is not the document in its entirety, but only relevant document fragments and thus, granularity of the search terms must be refined when searching over XML document corpus. Second, the result of a keyword search over XML documents must be semantically interconnected document fragments. Finally, XML documents include large amounts of textual information and part of this is rarely searched. Building a single index for the whole document will make the index bulky and difficult to manage. The possibility of retrieval and ranking of XML fragments based on keyword queries using Adaptive GA for learning tag information is explored in (Srinivasa K G et al., 2005:ICP).

MAIN FOCUS

The application of soft computing paradigms like Genetic Algorithms and Support Vector Machines are used effectively to solve the optimization problems. The XML topic categorization is efficiently carried out using SVMs. Once the XML documents are categorized, the related and relevant information has to be disseminated to the user by a genetically learned user model in combination with SVMs. Once a large number of such XML tags exist, an efficient search over such XML repository is carried out using Adaptive GAs.

Topic Categorization Using SVM

The need for categorization of XML documents into specific user interest categories is in great demand because of huge XML repositories. A machine learning approach is applied to *topic categorization* which makes use of a multi class SVM for exploiting the semantic content of XML documents. The SVM is supplemented by a feature selection technique which is used to extract the useful features. For the application of SVM to the given XML data a feature vector must be constructed. The choice of the feature set determines the overall accuracy of the categorization task. Therefore, all the distinct tags from the training set XML documents are collected. This represents the initial tag pool. The tag pool can have rarely used tags and tags spread over all the categories apart from the more frequently used tags. Such tags can deteriorate the performance of SVM. The purpose of feature selection is to select the optimal feature subset that can achieve highest accuracy. Then, the XML document is parsed and all the tags present in the document are extracted. Only binary values are assigned as dimensions of the feature vector and supplied as the input to the multi class SVM. Later, this classifier is used for categorizing a new XML document based on the topic of relevance.

Selective Dissemination of XML Fragments

As the number of documents published in the form of XML is increasing, there is a need for selective dissemination of XML documents based on user interests. A combination of Adaptive Genetic Algorithms (Srinivasa K G et al., 2007:Elsevier) and a multi class SVM is used to learn a user model. Based on the feedback from the users, the system automatically adapts to the user's preference and interests. The user model and a similarity metric are used for selective dissemination of a continuous stream of XML documents. Using GAs to learn user profiles has two advantages. First, the tag combinations which are interesting to a user can be extracted using relevance feedback mechanism. Second, the context of the search terms given by the users can be adjudged and a profile can be constructed accordingly. From the collection of user profiles, random profiles are sampled and a decision on the category to which they belong is made.

A feature extractor and an SVM are used for the user model construction. The Feature extraction is performed using the measure of expected entropy loss to rank the features that are discriminators among the categories. An extended SVM to support multiclass classification is used to build the user model. A voting vector with a dimension for each class is also used for classifica-

tion. There are as many votes as the number of SVMs and the class having the maximum number of votes yields the corresponding support vectors. Here, SVM first classifies the various profiles into user interest category and then the same model is used to assign a user interest category to an incoming XML document from among the various pre-specified categories. After the user model has assigned a user interest category to the incoming XML document the next step is to find the users to whom the document is to be dispatched. A similarity metric between the incoming document and the user profile can judge the users to whom the document will be of most interest and hence the selective dissemination.

A Semantic Search Using Adaptive Genetic Algorithms

The XML document can be considered as a directed, node-labeled *data graph* $G = (X, E)$. Each node in X corresponds to an XML element in the document and is characterized by a unique *object identifier,* a *label* that captures the semantics of the element and leaf nodes are associated with a sequence of *keywords*. E is the set of edges which define the relationships between nodes in X. The edge $(l, k) \in E$, if there exists a directed edge from node l to node k in G. The edge $(l, k) \in E$ also denotes that node l is the *parent* of node k in G. Node l is also the ancestor of node k if a sequence of directed edges from node l leads to node k. Let the XML document tree be called τ. Let x be an interior node in this tree. We say that x directly satisfies a search term k if x has a leaf child that contains the keyword k and x indirectly satisfies a keyword k if some descendent of x directly satisfies the search term k. A search query $q = \{k_1, k_2, \ldots k_m\}$ is satisfied by a node x *iff* x satisfies each of $k_1, k_2, \ldots k_m$ either directly or indirectly. First a representative training set is chosen to assist the genetic learning of tags. The keyword queries and the relevant search results are collected from the user.

An adaptive genetic algorithm retrieves the tag combination which can answer a maximum number of training queries. Separate indices are built for the frequently used and occasionally used tag combinations. A search over the XML documents in the decreasing order of importance of tags is performed. Finally, the search produces only semantically related results. Making use of adaptive GAs to learn the tag information has two advantages. First, when the same keyword appears more than once in the XML document with different semantics (different tags), the knowledge learnt from the GA is used to rank the search results. Hence, the results that are more relevant to the user queries are better ranked than the other results. Second, separate indices can be built for both the frequently and less frequently searched tags. Thus the indices are smaller in size and hence manageable. The notations for relationship strength and semantic relationship help in efficient retrieval of semantically interconnected results as well as ranking the search results based on the proximity of the keywords.

FUTURE TRENDS

The future of the WWW is the semantic web. Since XML is the integral part of web semantics, mining such XML repository is very essential for knowledge dissemination. There are several possible avenues for future extensions. The algorithms can be refined such that they are able to cluster the users with similar interests together. A system which allows users to explicitly specify their preferences apart from automatically learning their interests will be very efficient.

CONCLUSION

A framework for topic categorization of XML documents that makes use of SVM for the purpose of classification is discussed. Feature selection schemes that can improve the accuracy of the SVM is also addressed. The All-Vs-One Multiclass SVM is used to assign a topic category to the input XML document from among the various pre-specified categories. A system for Selective dissemination of XML documents that makes use of genetic algorithms to learn user interests is discussed. A model which assigns user interest categories to the user profiles as well as the streaming XML documents is also discussed. Another framework for information retrieval from XML documents that uses tag information to improve the retrieval performance is explored. Adaptive GAs, which are efficient for search in large problem spaces, are used to learn the significance of the tags.

REFERENCES

Freitas, Alex A. (2002). Data mining and knowledge discovery with evolutionary algorithms. *Natural Computing Series, Springer.*

Gang Gou and Rada Chirkova. (2007). Efficient querying large XML data repositories: A survey. *IEEE Transactions on Knowledge and Data Engineering, 19*(10), 1381-1403.

Luk, R., et al., (2000). A survey of search engines for XML documents. *SIGIR Workshop on XML and IR.*

Mitra, S., Pal, S. K., & Mitra, P. (2002). Data mining in soft computing framework: A survey. *IEEE Transactions on Neural Networks, 13*(1), 3-14.

Srinivasa, K. G., Sharath, S., Venugopal, K. R., & Patnaik, L. M. (2005). A framework for topic categorization of XML documents using support vector machines. *Innovative Applications of Information Technology for the Developing World,* (pp.367-371). UK: Imperial College Press.

Srinivasa, K. G., Venugopal, K. R., & Patnaik, L. M. (2007). Self adaptive migration model genetic algorithms. *Information Sciences, 177*(20), 4295-4313. Elsevier.

Srinivasa, K. G., Venugopal, K. R., & Patnaik, L. M. (2007). Selective dissemination of XML documents based on genetically learned user model and support vector machines. *International Journal on Intelligent Data Analysis, 11*(5), 481-496.

Srinivasa, K. G., Sharath, S., Venugopal, K. R., & Patnaik, L. M. (2005). An XML information retrieval mechanism using self-adaptive genetic algorithms. *International Journal of Computational Intelligence and Applications, 5*(4), 471-493. Imperial College Press.

KEY TERMS

Genetic Algorithms: Genetic algorithms (GAs) are search procedures that use the mechanics of natural selection and natural genetics.

Selective Dissemination of Information: SDI distributes automatically and electronically to end users the tables of contents from Web in which they are interested. **Semantic Web:** The Semantic Web is an extension of the current Web that will allow you to *find*, *share*, and *combine* information more easily.

Support Vector Machines: Algorithm used to identify patterns in datasets.

Topic Categorization: The use of computer software to categorise web pages based on a particular topic. Taxonomies for categorisation can also be created automatically.

User Modeling: Construction of (typically computer-based) models of users' mental activities and behaviors often used to make predictions about a system's usability or as a basis for interactive help systems.

Soft Subspace Clustering for High-Dimensional Data

Liping Jing
Hong Kong Baptist University, Hong Kong

Michael K. Ng
Hong Kong Baptist University, Hong Kong

Joshua Zhexue Huang
The University of Hong Kong, Hong Kong

INTRODUCTION

High dimensional data is a phenomenon in real-world data mining applications. Text data is a typical example. In text mining, a text document is viewed as a vector of terms whose dimension is equal to the total number of unique terms in a data set, which is usually in thousands. High dimensional data occurs in business as well. In retails, for example, to effectively manage supplier relationship, suppliers are often categorized according to their business behaviors (Zhang, Huang, Qian, Xu, & Jing, 2006). The supplier's behavior data is high dimensional, which contains thousands of attributes to describe the supplier's behaviors, including product items, ordered amounts, order frequencies, product quality and so forth. One more example is DNA microarray data.

Clustering high-dimensional data requires special treatment (Swanson, 1990; Jain, Murty, & Flynn, 1999; Cai, He, & Han, 2005; Kontaki, Papadopoulos & Manolopoulos., 2007), although various methods for clustering are available (Jain & Dubes, 1988). One type of clustering methods for high dimensional data is referred to as subspace clustering, aiming at finding clusters from subspaces instead of the entire data space. In a subspace clustering, each cluster is a set of objects identified by a subset of dimensions and different clusters are represented in different subsets of dimensions.

Soft subspace clustering considers that different dimensions make different contributions to the identification of objects in a cluster. It represents the importance of a dimension as a weight that can be treated as the degree of the dimension in contribution to the cluster. Soft subspace clustering can find the cluster memberships of objects and identify the subspace of each cluster in the same clustering process.

BACKGROUND

Finding clusters from subspaces of high dimensional data, subspace clustering pursues two tasks, identification of the subspaces where clusters can be found and discovery of the clusters from different subspaces, i.e., different subsets of dimensions. According to the ways with which the subsets of dimensions are identified, subspace clustering methods are divided into the following two categories. *Hard subspace clustering* determines the exact subsets of dimensions where clusters are discovered. Typical examples include PROCLUS, HARP and others. (Chakrabarti & Mehrotra, 2000; Yip, Cheung, & Ng, 2004 ; Parsons, Haque, & Liu, 2004). *Soft subspace clustering* considers that each dimension makes a different level of contribution to the discovery of clusters and the degree of contribution of a dimension to a cluster is represented as the weight of this dimension. The subsets of the dimensions with larger weights in a cluster form the subspace of the cluster. Typical examples include LAC, COSA, SCAD and others (Domeniconi, Papadopoulos, Gunopulos, & Ma, 2004; Frigui and Nasraoui, 2004; Friedman and Meulman, 2004; Chan, Ching, Ng, & Huang, 2004; Law, Figueiredo, & Jain, 2004).

The above subspace clustering methods have more or less three problems. Firstly, they are not scalable to large data (e.g., HARP, COSA). Large high dimensional data can not be well handled with them. Secondly, some use a projection method (e.g., PROCLUS), which makes the clustering results non-understandable.

Recovery of the original dimensions from the projected dimensions turns out to be difficult. Thirdly, some (e.g., SCAD, LAC) can not handle sparse data, which is a well-known phenomenon in real applications (Jing, Huang, & Ng, 2005).

MAIN FOCUS

This chapter is focused on a new soft subspace clustering method. This method determines the subspaces of clusters according to the contributions of the dimensions in discovering the corresponding clusters. The contribution of a dimension is measured by a weight that is assigned to the dimension in the clustering process. Every dimension contributes to the discovery of clusters, but the dimensions with larger weights identify the subspaces of the clusters (Jing, Ng, & Huang, 2007). The new soft subspace clustering algorithm is based on the *k*-means clustering process. Therefore, it can cluster large and high-dimensional sparse data.

The *k*-Means Algorithm

The *k*-means algorithm (MacQueen, 1967) is the mostly used clustering algorithm in data mining. Given a set of numeric objects X and an integer *k*, the *k*-means algorithm searches for a partition of X into *k* clusters that minimizes the sum of the within groups squared errors. This process is often formulated as the following minimization problem.

$$F(U,Z)=\sum_{l=1}^{k}\sum_{i=1}^{n}\sum_{j=1}^{m}u_{i,l}(x_{i,j}-z_{l,j})^2 \quad (1)$$

subject to

$$\sum_{l=1}^{k}u_{i,l}=1,\ 1\le i\le n,\ 1\le l\le k,\quad u_{i,l}\in\{0,1\}$$

where $U=[u_{i,l}]$ is an $n\times k$ partition matrix and $u_{i,l}$ = 1 indicates that the ith object is allocated to the lth cluster. $Z=[z_{l,j}]$ is a set of k vectors representing the centers of the k clusters.

Problem P can be solved by iteratively solving two sub minimization problems. One is to minimize $F(U,\hat{Z})$ with a given $\hat{Z}$ as constant by

$$u_{i,l}=1 \text{ , if } \sum_{i=1}^{m}(x_{i,j}-z_{l,j})^2\le\sum_{i=1}^{m}(x_{i,j}-z_{r,j})^2 \text{ for } 1\le r\le k, \quad (2)$$

$u_{i,l}$ = 0, otherwise.

The other is to minimize $F(\hat{U},Z)$ with a given partition matrix $\hat{U}$ by

$$z_{l,j}=\frac{\sum_{i=1}^{n}u_{i,l}x_{i,j}}{\sum_{i=1}^{n}u_{i,l}} \text{ for } 1\le l\le k, \quad \text{and} \quad 1\le j\le m. \quad (3)$$

The convergence of this k-means minimization process is proved in (Selim & Ismail, 1984).

One of the drawbacks of the *k*-means algorithm is that it treats all features equally in deciding the cluster memberships of objects. This is not desirable when dealing with high dimensional data with a large number of diverse features (dimensions). In such data, a cluster structure is often confined to a subset of features rather than the whole feature set. Inclusion of other features can only obscure discovery of the cluster structure in the clustering process. This drawback can be removed with a feature weighting technique that can identify the feature importance and help the *k*-means algorithm to find clusters in subset of features.

Feature Weights

Feature weighting for clustering is an important research topic in statistics and data mining (Modha & Spangler, 2003; Huang, Ng, Rong, & Li, 2005). The main purpose is to select important features in which a weight is assigned to a dimension for the entire data set. A weight can also be assigned to a feature in each cluster according to the feature importance in forming the corresponding cluster (Jing, Huang, & Ng, 2005; Jing, Ng, & Huang, 2007). In other words, a *k x m* weight matrix $V = [v_{l,j}]$ (l is the cluster index and j the feature index) is built in the clustering process. In this matrix, different clusters have different feature weight values. The weight value for a feature in a cluster is inversely proportional to the dispersion of the feature values from the center of the cluster. Therefore, the high weight value indicates a small dispersion of the feature values in the

cluster, which means that the feature is more important in forming the cluster. A cluster can be identified from the subspace with important features (large weights), and at the same time these important features can be used to describe the meaning of the cluster.

Objective Function

The new objective function is based on the objective function of the standard *k*-means by introducing the feature weights. A term of the entropy of the feature weights is also added. This entropy term represents the certainty of features in the identification of a cluster. Therefore, the clustering process simultaneously minimizes the within cluster dispersion and maximizes the negative weight entropy to stimulate more features to contribute to the identification of clusters. In this way, the problem of identifying clusters by few features in sparse data can be avoided.

The new objective function is given as

$$F(U,Z,V)=\sum_{l=1}^{k}\left[\sum_{i=1}^{n}\sum_{j=1}^{m}u_{i,l}v_{l,j}^{\beta}(x_{i,j}-z_{l,j})^2+\gamma\sum_{i=1}^{m}v_{l,j}\log v_{l,j}\right] \quad (4)$$

subject to

$$\sum_{l=1}^{k}u_{i,l}=1,\ 1\le i\le n,\ 1\le l\le k,\ u_{i,l}\in\{0,1\}$$

$$\sum_{i=1}^{m}v_{l,j}=1,\ 1\le j\le m,\ 1\le l\le k,\ 0\le v_{l,j}\le 1.$$

Here, $V=[v_{l,j}]$ is the $k\times m$ weight matrix, $\beta(>1)$ and γ are two given factors.

The new objective function contains two terms. The first term is the sum of the within cluster dispersions and the second term is the negative weight entropy. The parameter γ controls the strength of the incentive for clustering on more features.

Algorithm

The usual method toward optimization of the objective function (4) with the constraints is to use the partial optimization for one set of variables with other sets of variables treated as constants. The iterative minimization process is formed in three steps.

Step 1: Given $\hat{Z}$ and $\hat{V}$, update the partition matrix *U* by

$$u_{i,l}=1,\ \text{if}\ \sum_{j=1}^{m}v_{l,j}(x_{i,j}-z_{l,j})^2\le\sum_{i=1}^{m}v_{r,j}(x_{i,j}-z_{r,j})^2\ \text{for}\ 1\le r\le k, \quad (5)$$

$ui,l = 0$, otherwise

Step 2: Given $\hat{U}$ and $\hat{V}$, update the cluster centers Z with (3).

Step 3: Given $\hat{U}$ and $\hat{Z}$, calculate the feature weight matrix V as

$$v_{l,j}=\frac{\exp(-D_{l,j}/\gamma)}{\sum_{t=1}^{m}\exp(-D_{l,t}/\gamma)} \quad (6)$$

where

$$D_{l,t}=\sum_{i=1}^{n}u_{i,l}(x_{i,t}-z_{l,t})^2.$$

The soft subspace clustering algorithm is summarized as shown in Algorithm 1.

Algorithm 1 indicates that the new soft subspace clustering method only adds one new step to the *k*-means clustering process to calculate the feature weights of each cluster. Therefore, it is scalable to the number of features, the number of objects and the number of clusters.

FUTURE TRENDS

Similar to the drawbacks of the standard *k*-means, the soft subspace clustering algorithm is also sensitive to the initial settings, e.g., the number of clusters and the initial cluster centers. An effective subspace cluster validation method is needed to evaluate the clustering results. Some cluster validity index (Xie and Beni, 1991; Sun, Wang, & Jiang, 2004) may be modified to validate clusters in subspaces.

The current study of this algorithm has been focused on numeric data. However, a lot of data are categorical in many application domains. How to apply the soft

Algorithm 1:
Input: The number of clusters k and parameter ;
Randomly choose k cluster centers and set all initial weights to 1/m;

Repeat
  Update the partition matrix U by (6);
Update the cluster centers Z by (3);
Update the feature weights V by (5);
Until the objective function (4) obtains its local minimum value.

subspace clustering algorithm to high dimensional data is an interesting research problem that needs to be solved.

Furthermore, outlier detection should be considered in subspace clustering. The preferable way to deal with outliers in partitioning the data is to keep one extra set of outliers, so as not to pollute factual clusters. For instance, the algorithm CLIQUE (Agrawal, Gehrke, Gunopulos, & Raghavan, 1998) handles outliers by eliminating the subspaces with low coverage.

In addition, missing value is an actual challenging issue confronted by subspace clustering, also by data mining. Existing methods dealing with missing values can be roughly treated as a procedure that replaces the missing values in a dataset by some plausible values. The plausible values are generally generated from the dataset using a deterministic or random method. More efficient approach for clustering high dimensional data with missing values should be investigated. This is essentially a sparse data clustering problem.

CONCLUSION

The demand for clustering methods to handle high dimensional data will increase as more and more structured or unstructured data sources are available (e.g., text, image, audio, etc.). The soft subspace clustering algorithm based on the efficient k-means clustering process provides a useful technique for solving large high dimensional data clustering problems. Further study is needed to test this algorithm in different applications. One interesting research topic is validation of clusters discovered from subspaces. The existing cluster validation indices do not consider this situation.

REFERENCES

Agrawal, R., Gehrke, J., Gunopulos, D. & Raghavan, P. (1998), *Automatic subspace clustering of high dimensional data for data mining application*, Proc. of ACM SIGMOD, 94-105.

Cai, D., He, X. & Han, J. (2005), *Document clustering using locality preserving indexing*, IEEE transactions on knowledge and data engineering, 17(12), 1624-1637.

Chakrabarti, K. & Mehrotra, S. (2000), *Local dimensionality reduction: a new approach to indexing high dimensional spaces*, Proc. of 26th International conference of very large data bases, 89-100.

Chan, Y., Ching, W., Ng, M. K. & Huang, J. Z. (2004), *An optimization algorithm for clustering using weighted dissimilarity measures*, Pattern recognition, 37(5), 943-952.

Domeniconi, C., Papadopoulos, D., Gunopulos, D. & Ma, S. (2004), *Subspace clustering of high dimensional data*, Proc. of SIAM International conference of data mining.

Friedman, J. H. & Meulman, J. J. (2004), *Clustering objects on subsets of attributes*, Journal of royal statistical society B, 66(4), 815-849.

Frigui, H. & Nasraoui, O. (2004), U*nsupervised learning of prototypes and attribute weights*, Pattern recognition, 37(3), 567-581.

Huang, J. Z., Ng, M. K., Rong, H. & Li, Z. (2005), *Automated variable weighting in k-means type clustering*, IEEE transactions on pattern analysis and machine intelligence, 27(5), 1-12.

Jain, A. K. & Dubes, R. C. (1988), *Algorithms for clustering data*, Prentice-Hall, Englewood Cliffs, NJ, USA.

Jain, A. K., Murty, M N. & Flynn, P. L. (1999), *Data Clustering: a review*, ACM computing surveys, 31(3), 264-323.

Jing, L., Huang, J. Z. & Ng, M. K. (2005), *Subspace clustering of text documents with feature weighting k-means algorithm*, Proc. of the 9th Pacific-Asia conference on knowledge discovery and data mining, 802-812.

Jing, L., Ng, M. K. & Huang, J. Z. (2007), *An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data*, IEEE transactions on knowledge and data engineering, 19(8), 1026-1041.

Kontaki, M. Papadopoulos, A. N. & Manolopoulos, Y. (2007), *Continuous subspace clustering in streaming time series*, Information systems, to appear.

Law, M., Figueiredo, M. & Jain, A. K. (2004), *Simultaneous features selection and clustering using mixture models*, IEEE transaction on pattern analysis and machine intelligence, 26(9), 1-13.

MacQueen, J. B. (1967), *Some methods for classification and analysis of multivariate observations*, Proceedings of the 5th Berkeley symposium on mathematical statistics and probability, 281-297.

Modha, D. S. & Spangler, W. S. (2003), *Feature weighting in k-means clustering*, Machine learning, 52, 217-237.

Parsons, L., Haque, E. & Liu, H. (2004), *Subspace clustering for high-dimensional data: a review*, SIGKDD explorations, 6(1), 90-105.

Selim, S. & Ismail, M. (1984), *K-means-type algorithms: a generalized convergence theorem and characterization of local optimality*, IEEE transactions on pattern analysis and machine intelligence, 6(1), 81-87.

Sun, H., Wang, S. & Jiang, Q. (2004), *Fcm-based model selection algorithms for determining the number of clusters*. Pattern recognition, 37, 2027-2037.

Swanson, D.R. (1990), *Medical literature as a potential source of new knowledge*, Bull. Medical Library Assoc., 17(1), 29-37.

Xie, X. & Beni, G. (1991), *A validity measure for fuzzy clustering*. IEEE transaction on pattern analysis and machine intelligence, 13(8), 841-847.

Yip, K. Y., Cheung, D. W. & Ng, M. K. (2004), *A practical projected clustering algorithm*, IEEE transactions on knowledge and data engineering, 16(11), 1387-1397.

Zhang, X. , Huang, J. Z., Qian , D., Xu, J., & Jing, L. (2006). *Supplier categorization with k-means type subspace clustering*. In X. Zhou, J. Li, H. T. Shen, M. Kitsuregawa & Y. Zhang (Eds), APWEB2006. *Frontiers of WWW Research and Development* (pp. 227-237), LNCS 3841, Springer.

KEY TERMS

Cluster Validation: Process to evaluate a clustering result to determine whether the "true" clusters inherent in the data are found.

Clustering: A process of grouping data objects into clusters according to some similarity measure.

Feature Weighting: A technique to evaluate a feature with some criterion, e.g., its importance and then assign a value to a feature.

High Dimension Data: The data with hundreds or thousands of dimensions, for example, gene or text data.

k-Means: A partitional clustering technique that partitions data objects into *k* clusters by iteratively minimizing the sum of the within cluster dispersions.

Subspace Clustering: An extension of the traditional clustering method by finding clusters in different subspaces formed by a subset of features.

Soft Subspace Clustering: The soft subspace clustering approach determines the subsets of dimensions according to contributions of the features in discovering the corresponding clusters, and the contribution of a feature dimension is measured by a weight that is assigned to the feature in the clustering process.

Spatio-Temporal Data Mining for Air Pollution Problems

Seoung Bum Kim
The University of Texas at Arlington, USA

Chivalai Temiyasathit
The University of Texas at Arlington, USA

Sun-Kyoung Park
North Central Texas Council of Governments, USA

Victoria C. P. Chen
The University of Texas at Arlington, USA

INTRODUCTION

Vast amounts of data are being generated to extract implicit patterns of ambient air pollution. Because air pollution data are generally collected in a wide area of interest over a relatively long period, such analyses should take into account both temporal and spatial characteristics. Furthermore, combinations of observations from multiple monitoring stations, each with a large number of serially correlated values, lead to a situation that poses a great challenge to analytical and computational capabilities. Data mining methods are efficient for analyzing such large and complicated data. Despite the great potential of applying data mining methods to such complicated air pollution data, the appropriate methods remain premature and insufficient. The major aim of this chapter is to present some data mining methods, along with the real data, as a tool for analyzing the complex behavior of ambient air pollutants.

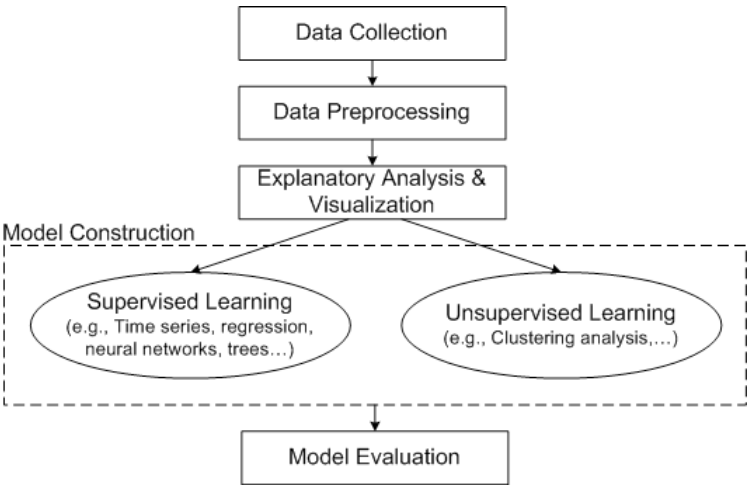
BACKGROUND

In 1990, under the Clean Air Act., the U.S. Environmental Protection Agency (EPA) set the National Ambient Air Quality Standards (NAAQS) for six pollutants, also known as criteria pollutants, which are particulate matter, ozone, sulfur dioxide, nitrogen dioxides, carbon monoxide, and lead (US EPA, 1990). Any exceedance of the NAAQS results in non-attainment of the region for that particular pollutant.

Well-known consequences of air pollution include the green house effect (global warming), stratospheric ozone depletion, tropospheric (ground-level) ozone, and acid rain (Wark, Warner, & Davis, 1998). In this chapter, we present applications on tropospheric ozone and the less publicized air pollution problem of particulate matter. High concentrations of tropospheric ozone affect human health by causing acute respiratory problems, chest pain, coughing, throat irritation, or even asthma (Lippmann, 1989). Ozone also interferes with the ability of plants to produce and store food, damages the leaves of trees, reduces crop yields, and impacts species diversity in ecosystems (Bobbink, 1998; Chameides & Kasibhatla, 1994). Particulate matter is an air contaminant that results from various particle emissions. For example, $PM_{2.5}$ (particulate matter that is 2.5 micrometers or smaller in size) has the potential to cause adverse health effects in humans, including premature mortality, nose and throat irritation, and lung damage (e.g., Pope et al., 2002). Furthermore, $PM_{2.5}$ has been associated with visibility impairment, acid deposition, and regional climate change.

To reduce pollutant concentrations and establish the relevant pollution control program, a clear understanding of the pattern of pollutants in particular regions and time periods is necessary. Data mining techniques can help investigate the behavior of ambient air pollutants and allow us to extract implicit and potentially useful knowledge from complex air quality data. Figure 1 illustrates the five primary stages in the data mining process in air pollution problems: data collection, data preprocessing, explanatory analysis and visualization, model construction, and model evaluation.

Figure 1. Overview of data mining in air pollution problems.



MAIN FOCUS OF CHAPTER

Data Collection

Because air pollution data are generally collected in a wide region of interest over a relatively long time period, the data are composed of both temporal and spatial information. A typical air pollution database consists of pollutant observations $O(S_i, T_j)$, for monitoring site S_i at time T_j for $i=1,2,\ldots,m$, $j=1,2,\ldots,n$, where m and n is the number of monitoring sites and time points, respectively. Since most air pollution data hold these two properties, spatial and temporal variability should be incorporated into the analysis in order to accurately analyze the air pollution characteristics. Table 1 provides a list of publicly accessible databases and their web addresses that contain a variety of air pollution data.

Data Preprocessing

Preprocessing of air pollution data is a crucial task because inadequate preprocessing can result in low-quality data and make it difficult to extract meaningful information from subsequent analyses. The collected air pollution data typically contain a number of potential outliers that are far away from the rest of the observations and missing values possibly due to measurement or instrumental errors. It is necessary to process missing values and outliers in both the time and space domains. Imputing missing values or replacing potential outliers with a sample average is the simplest method because it can be calculated without any pre-specified assumptions or complex mathematical formulas. However, the sample average assumes that each observation is equally important and does not take into account the fact that the data are collected over time and space. A weighted average can be an efficient method to replace the outliers or impute the missing values. One example of using a weighted average is the inverse-distance-squared weighted method (McNair, Harley, & Russell, 1996). This method determines weights based on spatial proximity to the query points. The interpolated value for site S_i at time T_j, $I(S_i, T_j)$ is computed as follows:

$$I(S_i, T_j) = \frac{\sum_{k=1,k\neq i}^{m} O(S_k, T_j) \cdot \omega_k}{\sum_{k=1,k\neq i}^{m} \omega_k}, \quad (1)$$

where m is the number of monitoring sites and ω_s is calculated as follows:

$$\omega_k = \begin{cases} \frac{1}{r_k^2} & if \quad r_k \leq d \\ 0 & if \quad r_k > d \end{cases} \quad (2)$$

Table 1. Example of publicity accessible database for air pollution data

Network	Locations	Parameters	Time Range	Sources
AIRS-Gaseous	United States (U.S.)	O_3, CO, SO_2, NO_2, PM Mass concentrations	1990-present	AIRS/AQS http://www.epa.gov/ttn/airs/airsaqs/
PAMS, AIRS-Gaseous	U.S. Ozone Nonattainment area	O_3, NO_2, NO_X, Nitric Acid	1994-present	Same as AIRS-Gaseous
AIRS-Speciated	U.S.	$PM_{2.5}$ Mass Concentration, Speciated Aerosol	2001-present	Same as AIRS-Gaseous
SEARCH - Continuous	Southeastern U.S.	$PM_{2.5}$ Mass Concentration, Speciated Aerosol, Gaseous, Surface Meteorology	1998-present	http://www.atmospheric-research.com/public/index.html
SEARCH - 24 hour	Southeastern U.S.	$PM_{2.5}$ Mass Concentration, Speciated Aerosol	1998-present	Same as SEARCH-Continuous

r_k is the distance from site S_i to site S_k at time T_j and d can be specified by the users. Thus, $I(S_i, T_j)$ is the weighted average value observed in the surrounding m sites in which the weights are determined by the way that observations in close spatial proximity are given more weight than those that are spatially separated.

Other approaches for processing outliers and missing values include functional or maximum likelihood imputation schemes. Polynomial functions and splines can be used to interpolate regularly-spaced data. Maximum likelihood, which typically requires high computation, uses an iterative approach based on model parameter estimation. Examples of this approach include Expectation-Maximization (Schafer, 1997) and kriging (Stein, 2006).

Explanatory Analysis and Visualization

The main purpose of exploratory analysis and visualization is to provide initial guidelines that enable the subsequent analyses to be more efficient. Principal component analysis (PCA) is a multivariate data analysis technique that helps reduce the dimension of a data set via an orthogonal linear transformation (Jolliffe, 2002). The transformed variables, called principal components (PCs) are uncorrelated, and generally, the first few PCs are sufficient to account for most of variability of the entire data. Thus, plotting the observations with these reduced dimensions facilitates the visualization of high-dimensional data. PCA has been used in a variety of air pollution applications, for example, Tilmes & Zimmermann (1998) and Abdul-Wahab & Al-Alawi (2002). Lengyel et al. (2004) observed the diurnal pattern (day and night) of tropospheric ozone concentrations using reduced dimensions in PCA. Lehman et al. (2004) applied a rotated PCA approach to study the spatial and temporal variability of tropospheric ozone concentrations in the eastern United States.

Correspondence analysis is another useful explanatory technique that analyzes the relationship between two or more categorical variables. Correspondence analysis examines the contingency table containing the frequency data to investigate how it deviates from expectation assuming the columns and rows are independent (Johnson & Wichern, 2002). Similar to PCA, correspondence analysis provides low-dimensional data that facilitate the visualization of the association in multiple levels in contingency tables. Multiple cor-

respondence analysis that extends to the case of more than three categorical variables was used to examine the relationship between nitrogen dioxide exposure levels and related qualitative variables (Piechocki-Minguy et al., 2006).

Model Construction

Data mining tools for constructing models can be divided into two categories, supervised and unsupervised approaches. Supervised approaches require both the explanatory variable and the response variable, while unsupervised approaches rely solely upon the explanatory variables. Time series analysis is one of the classical supervised approaches for analyzing the data, collected over time. Numerous studies have used time-series analysis to investigate and predict the behavior of air pollution (Lehman et al., 2004; Shi & Harrison, 1997). Recently, Chelani and Devotta (2006) proposed a hybrid autoregressive integrated moving average (ARIMA) model that combined the Box and Jenkins ARIMA model with nonlinear dynamical modeling to forecast nitrogen dioxide concentrations.

Regression analyses aim to build the models based on the relationship between the explanatory and response variables. Regression analysis has been applied to identify the representative monitoring locations (Goswami, Larson, Lurnley, & Liu, 2002), and to predict a variety of air pollutant concentrations (Davis & Speckman, 1999; Lengyel et al., 2004). Artificial neural networks have also been widely used for predicting ozone concentrations in different locations around the world (e.g., Abdul-Wahab & Al-Alawi, 2002; Wang, Lu, Wang, & Leung, 2003).

Unsupervised approaches aim to extract the information purely from the explanatory variables. Although visualization techniques elicit the natural groupings of the observations, the interpretation of graphical results is not necessarily straightforward. Clustering analysis is an unsupervised approach that systematically partitions the observations by minimizing within-group variations and maximizing between-group variations, then assigning a cluster label to each observation. Numerous clustering methods have been introduced for grouping air pollution data (Gramsch, Cereceda-Balic, Oyola, & Von Baer, 2006; Oanh, Chutimon, Ekbodin,

Table 2. Performance measurement for model

Performance Measurement	Equation
Mean Biased (mg/m³)	$MB = \frac{1}{m*n}\sum_{j=1}^{n}\sum_{i=1}^{m}\left[O_f(S_i,T_j) - O_a(S_i,T_j)\right]$
Mean Error (mg/m³)	$MB = \frac{1}{m*n}\sum_{j=1}^{n}\sum_{i=1}^{m}\left\|O_f(S_i,T_j) - O_a(S_i,T_j)\right\|$
Root Mean Square Error (mg/m³)	$RMSE = \sqrt{\frac{1}{m*n}\sum_{j=1}^{n}\sum_{i=1}^{m}\left[\left(O_f(S_i,T_j) - O_a(S_i,T_j)\right)^2\right]}$
Mean Normalized Bias (%)	$MNB = \frac{1}{m*n}\sum_{j=1}^{n}\sum_{i=1}^{m}\left[\frac{O_f(S_i,T_j) - O_a(S_i,T_j)}{O_a(S_i,T_j)}\right]$
Mean Normalized Error (%)	$MNE = \frac{1}{m*n}\sum_{j=1}^{n}\sum_{i=1}^{m}\left\|\frac{O_f(S_i,T_j) - O_a(S_i,T_j)}{O_a(S_i,T_j)}\right\|$
Root Mean Square Normalized Error (%)	$RMSNE = \sqrt{\frac{1}{m*n}\sum_{j=1}^{n}\sum_{i=1}^{m}\left[\left(\frac{O_f(S_i,T_j) - O_a(S_i,T_j)}{O_a(S_i,T_j)}\right)^2\right]}$

$O_f(S_i, T_j)$: Fitted value from the model, $O_a(S_i, T_j)$: Actual value, m: Total number of monitoring sites, n: Total number of time points.

& Supat, 2005); however, no consensus exists about the best method to satisfy all conditions. A previous study applied the *k*-means clustering algorithm with Euclidean distance to sulfur dioxide data from 30 sites in the eastern United States. They obtained six clusters in which the sites within a cluster have a similar pattern of meteorological factors and sulfur dioxide levels (Holland, Principe, & Sickles, 1999).

Model Evaluation

The significance of constructed models should be evaluated for predicting the future behavior of air pollution. The basic approach for model evaluation is to separate data into two data sets, a training set and a testing set. The training set is used to construct the models and these models are then evaluated by their prediction ability on the testing set. Prediction errors typically measure the difference between the actual and fitted values. Table 2 lists the frequently used model performance evaluation measures for air pollution modeling.

CASE STUDIES

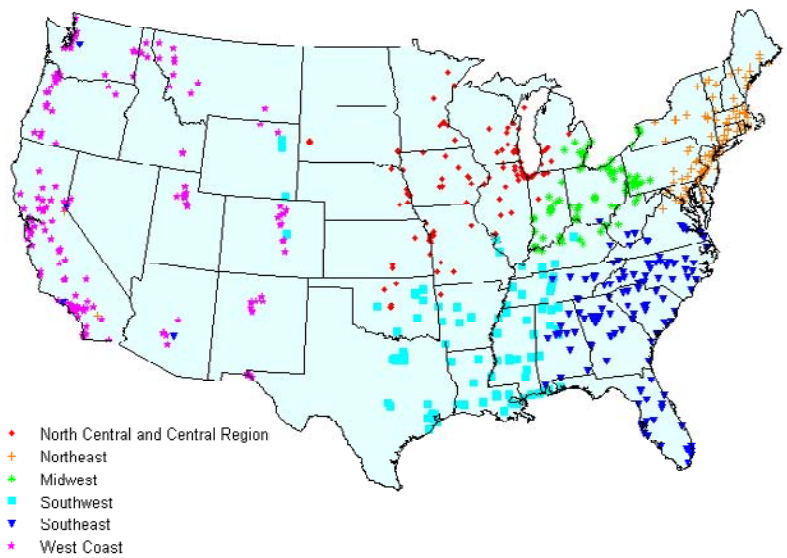
The main purpose of the case studies is to demonstrate how five stages of data mining process described above can be applied in real air pollution problems through the following two small examples.

Clustering Analysis of $PM_{2.5}$ Concentrations over the Continental United States

The purpose of this case study is to characterize spatial variations of $PM_{2.5}$ concentrations at 789 monitoring sites across the continental United States based solely upon their temporal patterns. Monitoring data were obtained from the Environmental Protection Agency's Aerometric Information Retrieval System (EPA-AIRS) database, which contains 24-hour average $PM_{2.5}$ concentrations measured every third day during 2000 (from January 2000 to December 2000) at 789 monitoring sites in the continental United States. Outliers and missing values were imputed in the data preprocessing step using the inverse-distance-squared weighted method (McNair et al., 1996).

A *k*-means clustering method was used to characterize the spatial variations of $PM_{2.5}$ concentrations based upon their temporal patterns. The results of the *k*-means clustering algorithm depend on the distance metrics and the number of clusters (*k*). For the distance metric, we used the correlation distance that allows us to measure the similarity in shape between the two temporal profiles

Figure 2. k-means clustering results (k=6) for the continental United States



from each monitoring site. We denote $Z(S_i, T_j)$ by the $PM_{2.5}$ concentration for monitoring site S_i at time T_j. Then, for the monitoring sites *x* and *y*, the correlation distance of two temporal profiles that consist of a series of *J* time points can be computed as follows:

$$D_{[Z(S_x,T_j),Z(S_y,T_j)]} = \frac{1}{J}\sum_{i=1}^{J}\left(\frac{Z(S_x,T_j)-\overline{Z}_{s_x}}{\sigma_{Z_{s_x}}}\right)\left(\frac{Z(S_y,T_j)-\overline{Z}_{s_y}}{\sigma_{Z_{s_y}}}\right), \quad (3)$$

where

$$\overline{Z}_{s_x} = \frac{1}{J}\sum_{j=1}^{J} Z(S_x,T_j) \text{ and } \sigma_{Z_{Sx}} = \left(\frac{1}{J}\sum_{j=1}^{J}\left(Z(S_x,T_j)-\overline{Z}_{s_x}\right)^2\right)^{1/2}. \quad (4)$$

To determine *k*, we used a heuristic approach that minimizes the variability within the same group and found the optimal number for *k* was six. The results of six-means clustering analysis are displayed on the U.S. map (Figure 2). It is seen that the monitoring sites in close spatial proximity are grouped together, demonstrating the identification of spatially homogeneous regions based on the temporal patterns of $PM_{2.5}$ concentrations.

The clustered sites can be grouped according to the following ad-hoc categories chosen by geographical locations, with the number of monitoring sites in each cluster indicated in parentheses: (1) North Central & Central Region (145); (2) Northeast (137); (3) Midwest (106); (4) Southwest (101); (5) Southeast (139); and (6) West (161).

Prediction of Ozone Concentration in the Dallas-Fort Worth Area

This case study attempts to investigate the behavior of daily maximum 8-hour ozone concentrations measured at 15 monitoring sites in the Dallas-Fort Worth (DFW) area from June 1, 2002 to May 31, 2006. Regression trees were developed to study how meteorological variables impact ozone concentrations. Monitoring data were obtained from the database maintained by the Texas Commission on Environmental Quality (TCEQ; http://www.tceq.state.tx.us), which contains daily maximum 8-hour ozone concentrations and meteorological variables from June 1, 2002 to May 31, 2006 at 15 monitoring sites in the DFW area. Missing values were imputed using inverse-distance-squared weighted method. The explanatory variables in a tree model include maximum wind gust, outdoor temperature, resultant wind direction, resultant wind speed, solar radiation, standard deviation of horizontal direction, wind speed, resultant wind speed from south to north, resultant wind speed from west to east, standard deviation of wind from south to north, and standard deviation of wind from west to east.

Regression tree analyses were conducted using the Classification and Regression Tree (CART) software (http://www.salfordsystems.com/). The actual derived tree is not shown here due to its size (the number of terminal nodes = 15). As an example, we obtained the following rule from one of the terminal nodes in the derived regression tree: "If the solar radiation was higher than 0.23 langleys per minute, ozone concentration lag 1 greater than 44.23 ppb, and speed less than 5.13 mph, the average ozone concentration is 64.292ppb (with standard deviations 10.244)." CART software provides "variable importance scores." The variable that receives a 100 score indicates the most influential variable for prediction, followed by other variables based on their relative importance to the most important one. The result shows that the ozone with single time lag, which addresses the serial correlation in ozone, is most important, followed by the solar radiation variables. It is interesting to note that for the solar radiation and temperature variables, both the single time lag and no time lag are important. This indicates a time-lagged effect of these variables affects the current ozone concentration. In other words, ozone concentration in the current day is impacted by the solar radiation and temperature in the previous day.

FUTURE TRENDS

Statistical analyses for time series or spatial data have been widely used to investigate the behavior of ambient air pollutants. Because air pollution data are generally collected over time at monitoring stations of interest, the analysis should simultaneously consider both spatial and temporal information. Despite numerous analytical tools available to analyze spatial or time series data separately, the appropriate procedures for analysis of spatio-temporal data are still lacking (Gryparis, Coull, Schwartz, & Suh, 2007; Schabenberger &

Gotway, 2005). Various data mining techniques can play an important role to identify inherent patterns in spatio-temporal data and help establish reliable models to understand the behavior of ambient air pollution. Further, the optimization of control strategies to reduce air pollution will rely on efficient and accurate models relating how the relevant variables change over time and space; thus, the role of data mining has significant potential for improving air quality in the future (Yang, Chen, Chang, Murphy, & Tsai, 2007).

CONCLUSION

Data mining techniques provide efficient approaches for investigating the behavior of ambient air pollution problems. This chapter provides an overview of the data mining process in air pollution modeling. This data mining process could also be applied to other application areas with spatial and temporal monitoring data, such as process control on semiconductor, water quality, meteorological studies, and airborne bioterrorism. Data mining applications along with two real case studies were discussed. Further analysis tools that consider spatial and temporal properties simultaneously are still needed.

REFERENCES

Abdul-Wahab, S. A., & Al-Alawi, S. M. (2002). Assessment and prediction of tropospheric ozone concentration levels using artificial neural networks *Environmental Modelling and Software, 17*(3), 219-228.

Bobbink, R. (1998). Impacts of tropospheric ozone and airborne nitrogenous pollutants on nature and semi-nature ecosystems: a commentary. *New Phytologist, 139*, 161-168.

Chameides, W. L., & Kasibhatla, P. S. (1994). Growth of continental-scale metro-agro-plexes, regional ozone pollution, and world food production. *Science, 264*(5155), 74-77.

Chelani, A. B., & Devotta, S. (2006). Air quality forecasting using a hybrid autoregressive and nonlinear model. *Atmospheric Environment, 40*(10), 1774-1780.

Davis, J. M., & Speckman, P. (1999). A model for predicting maximum and 8h average ozone in Houston. *Atmospheric Environment, 33*, 2487-2500.

Goswami, E., Larson, T., Lurnley, T., & Liu, L. J. S. (2002). Spatial Characteristics of Fine Particulate Matter: Identifying Representative Monitoring Location in Seattle, Washington. *Journal of the Air & Waste Management Association, 52*, 324-333.

Gramsch, E., Cereceda-Balic, F., Oyola, P., & Von Baer, D. (2006). Examination of pollution trends in Santiago de Chile with cluster analysis of PM10 and Ozone data. *Atmospheric Environment, 40*(28), 5464-5475.

Gryparis, A., Coull, B. A., Schwartz, J., & Suh, H. H. (2007). Semiparametric latent variable regression models for spatiotemporal modelling of mobile source particles in the greater Boston area. *Journal of the Royal Statistical Society: Series C (Applied Statistics), 56*(2), 183–209.

Holland, D. M., Principe, P. P., & Sickles, J. E. (1999). Trends in atmospheric sulfur and nitrogen species in the eastern United States for 1989-1995. *Atmospheric Environment, 33*, 37-49.

Johnson, R. A., & Wichern, D. W. (2002). *Applied Multivariate Statistical Analysis* (Fifth Edition ed.). Upper Saddle River, NJ: Prentice Hall.

Jolliffe, I. T. (2002). *Principal Component Analysis* (Second Edition ed.). New York, NY: Springer.

Lehman, J., Swinton, K., Bortnick, S., Hamilton, C., Baldridge, E., Eder, B., et al. (2004). Spatio-temporal characterization of tropospheric ozone across the eastern United States *Atmospheric Environment, 38*(26), 4357-4369.

Lengyel, A., Heberger, K., Paksy, L., Banhid, i. O., & Rajko, R. (2004). Prediction of ozone concentration in ambient air using multivariate methods. *Chemosphere, 57*(8), 889-896.

Lippmann, M. (1989). Health effects of ozone. A critical review. *Journal of Air Pollution Control Association, 39*(5), 672-675.

McNair, L. A., Harley, R. A., & Russell, A. G. (1996). Spatial inhomogeneity in pollutant concentrations, and their implications for air quality model evaluation. *Atmospheric Environment, 30*, 4291-4301.

Oanh, N. T. K., Chutimon, P., Ekbodin, W., & Supat, W. (2005). Meteorological pattern classification and application for forecasting air pollution episode potential in a mountain-valley area. *Atmospheric Environment, 39*(7), 1211-1225.

Piechocki-Minguy, A., Plaisance, H., Schadkowski, C., Sagnier, I., Saison, J. Y., Galloo, J. C., et al. (2006). A case study of personal exposure to nitrogen dioxide using a new high sensitive diffusive sampler. *Science of the Total Environment, 366*(1), 55-64.

Pope, C. A., Burnett, R., Thun, N. J., Calle, E. E., Krewskik, D., Ito, K., et al. (2002). Lung cancer, cardiopulmonary mortality, and long term exposure to fine particulate air pollution. *Journal of the American Medical Association, 287*, 1132-1141.

Schabenberger, O., & Gotway, C. A. (2005). *Statistical Methods for Spatial Data Analysis*. NY: Chapman & Hall.

Schafer, J. L. (1997). *Analysis of Incomplete Multivariate Data*. Boca Raton, FL: Chapman & Hall/CRC.

Shi, J. P., & Harrison, R. M. (1997). Regression modeling of hourly NO_x and NO_2 concentrations in urban air in London *Atmospheric Environment, 31*(24), 4081–4094.

Stein, M. L. (2006). *Interpolation of Spatial Data: Some Theory for Kriging*: Springer.

Tilmes, S., & Zimmermann, J. (1998). Investigation on the spatial scales of the variability in measured near-ground ozone mixing ratios *Geophysical Research Letters, 25*(20), 3827-3830.

US EPA. (1990). *Clean Air Act*. 2006, from http://www.epa.gov/oar/caa/caa.txt

Wang, W., Lu, W., Wang, X., & Leung, A. Y. (2003). Prediction of maximum daily ozone level using combined neural network and statistical characteristics. *Environment International, 29*(5), 555-562.

Wark, K., Warner, C. F., & Davis, W. T. (1998). *Air Pollution, Its Origin and Control*. Melon Park, CA: Addison-Wesley.

Yang, Z., Chen, V. C. P., Chang, M. E., Murphy, T. E., & Tsai, J. C. C. (2007). Mining and Modeling for a Metropolitan Atlanta Ozone Pollution Decision-Making Framework. *IIE Transactions, Special Issue on Data Mining, 39(6),* 607-615.

KEY TERMS

Clustering Analysis: Clustering analysis systematically partitions the observations by minimizing within-group variations and maximizing between-group variations, then assigning a cluster label to each observation.

Principal Component Analysis (PCA): PCA is a multivariate data analysis technique primarily for dimensional reduction and visualization. PCA creates new variables based on orthogonal transformations of the original variables.

Regression Tree: Regression trees partition the input variable space into disjoint hyper-rectangular regions and fit a model that predicts the response variable with a constant value.

Spatial Analysis: The systematic approach to understand the nature of geographic data.

Supervised Learning Approach: The process of establishing models based on both the input and output variables.

Time-Series Analysis: The statistical analysis to understand a sequence of data points collected over time and predicts future events.

Unsupervised Learning Approach: A modeling process that depends only on input variables but does not take into account the information from the response variable.

Spectral Methods for Data Clustering

Wenyuan Li
Nanyang Technological University, Singapore

Wee Keong Ng
Nanyang Technological University, Singapore

INTRODUCTION

With the rapid growth of the World Wide Web and the capacity of digital data storage, tremendous amount of data are generated daily from business and engineering to the Internet and science. The Internet, financial real-time data, hyperspectral imagery, and DNA microarrays are just a few of the common sources that feed torrential streams of data into scientific and business databases worldwide. Compared to statistical data sets with small size and low dimensionality, traditional clustering techniques are challenged by such unprecedented high volume, high dimensionality complex data. To meet these challenges, many new clustering algorithms have been proposed in the area of data mining (Han & Kambr, 2001).

Spectral techniques have proven useful and effective in a variety of data mining and information retrieval applications where massive amount of real-life data is available (Deerwester et al., 1990; Kleinberg, 1998; Lawrence et al., 1999; Azar et al., 2001). In recent years, a class of promising and increasingly popular approaches — spectral methods — has been proposed in the context of clustering task (Shi & Malik, 2000; Kannan et al., 2000; Meila & Shi, 2001; Ng et al., 2001). Spectral methods have the following reasons to be an attractive approach to clustering problem:

- Spectral approaches to the clustering problem offer the potential for dramatic improvements in efficiency and accuracy relative to traditional iterative or greedy algorithms. They do not intrinsically suffer from the problem of local optima.
- Numerical methods for spectral computations are extremely mature and well understood, allowing clustering algorithms to benefit from a long history of implementation efficiencies in other fields (Golub & Loan, 1996).
- Components in spectral methods have the naturally close relationship with graphs (Chung, 1997). This characteristic provides an intuitive and semantic understanding of elements in spectral methods. It is important when the data is graph-based, such as links of WWW, or can be converted to graphs.

In this paper, we systematically discuss applications of spectral methods to data clustering.

BACKGROUND

To begin with the introduction of spectral methods, we first present the basic foundations that are necessary to understand spectral methods.

Mathematical Foundations

Data is typically represented as a set of vectors in a high-dimensional space. It is often referred as the matrix representation of the data. Two widely used spectral operations are defined on the matrix.

- EIG(A) operation: Given a real symmetric matrix $A_{n\times n}$, if there is a vector $x \in R^n \neq 0$ such that $Ax=\lambda x$ for some scalar λ, then λ is called the eigenvalue of A with corresponding (right) eigenvector x. EIG(A) is an operation to compute all eigenvalues and corresponding eigenvectors of A. All eigenvalues and eigenvectors are real, that is, guaranteed by Theorem of real schur decomposition (Golub & Loan, 1996).
- SVD(A) operation: Given a real matrix $A_{m\times n}$, similarly, there always exists two orthogonal matrices $U \in R^{m\times m}$ and $V \in R^{n\times n}$ ($U^TU= I$ and $V^TV=I$) to decompose A to the form $A=USV^T$, where $S = \text{diag}(\sigma_1, ..., \sigma_r) \in R^{r\times r}$, r=rank(A) and $\sigma_1 \geq \sigma_2 ... \geq \sigma_r = ... = \sigma_n = 0$. Here, the σ_i are the singular values of A and the first r columns of U and V are the left and right (respectively) singular vectors of A. SVD(A) is called Singular Value Decomposition of A (Golub & Loan, 1996).

Typically, the set of eigenvalues (or singular values) is called the spectrum of A. Besides, eigenvectors (or

singular vectors) are the other important components of spectral methods. These two spectral components have been widely used in various disciplines and adopted to analyze the key encoding information of a complex system. Therefore, they are also the principle objects in spectral methods for data clustering.

Transformations

As observed by researchers, two key components of spectral methods — eigenvalues and eigenvectors — scale with different matrices. Therefore, before the analysis and application of them, some transformations, or more exactly, normalizations of two spectral components are needed. Although this might look a little complicated at first, this way to use them is more consistent with spectral geometry and stochastic processes. Moreover, another advantage of normalized components is due to its better relationship with graph invariants while the raw components may fail to do. There are three typical transformations often used in spectral methods.

- Laplacian: Given a symmetric matrix $A=(a_{ij})_{n\times n}$ with $a_{ij}\geq 0$, we define Laplacian $L_A=(l_{ij})_{n\times n}$ of A as

$$l_{ij}=\begin{cases}1-\dfrac{a_{ij}}{d_i}. & \text{if } i=j\\ -\dfrac{a_{ij}}{\sqrt{d_i d_j}}. & \text{if } a_{ij}\neq 0\\ 0. & \text{otherwise}\end{cases}$$

  The spectral graph theory takes this transformation (Chung, 1997).
- Variant of Laplacian: Given a symmetric matrix $A=(a_{ij})_{n\times n}$ with $a_{ij}\geq 0$, we define the variant of Laplacian $T_A=(t_{ij})_{n\times n}$ of A to be $T_A=D^{-1/2}(S-I)D^{-1/2}$. It can be easily proved that $L_A+T_A=2I$. This transformation of the matrix is often used (Li et al., 2004; Ng et al., 2001).
- Transition (or Stochastic) Matrix: Given a symmetric matrix $A=(a_{ij})_{n\times n}$ with $a_{ij}\geq 0$, we define the transition matrix $P_A=(p_{ij})_{n\times n}$ of A satisfying $p_{ij}=a_{ij}/d_i$ so that the sum of each row is 1. Apparently, P is a stochastic matrix, in the sense that it describes the transition probabilities of a Markov chain in the natural way.

In the definitions of these three matrices, $d_i=\sum_j a_{ij}$ is the sum of the i-th row vector and $D=\mathrm{diag}(d_1, \ldots, d_n)$. These three matrices have real eigenvalues and eigenvectors. Moreover, the eigenvalues of Laplacian and the transition matrix lie in [0, 2] and [-1, 1], respectively. We can easily deduce from the relationship between L_A and T_A to obtain $\mathrm{SPECTRUM}(T_A)=\{1-\lambda \mid \lambda\in \mathrm{SPECTRUM}(L_A)\}$, where SPECTRUM(•) represents the set of eigenvalues of a matrix. Hence, the eigenvalues of T_A lie in [-1, 1] and all the conclusions and properties of L_A are also applicable to T_A. Moreover, L_A and T_A have the same eigenvectors.

Relations to Graph

As a graph is represented by its adjacency matrix, there is a close relationship between the graph and the spectral components of its adjacency matrix. It is a long history to explore the fundamental properties of a graph from the view of the spectral components of this graph's adjacency matrix in the area of mathematics. Especially, eigenvalues are closely related to almost all major invariants of graphs and thus, play a central role in the fundamental understanding of graphs. Based on this perspective, spectral graph theory has emerged and rapidly grown in recent years (Chung, 1997). Hence, many characteristics of spectral components of a matrix can be intuitively explained in terms of graphs and meanwhile graphs also can be analyzed from its spectral components. A notable case is the authority and hub vertices of the Web graph that is important to Web search as shown in HITS algorithm (Kleinberg, 1998). Another example is that the spectrum of the adjacency matrix of a graph can be analyzed to deduce its principal properties and structure, including the optimization information about cutting a graph. This view has been applied to discover and predict the clustering behavior of a similarity matrix before the actual clustering is performed (Li et al., 2004).

MAIN THRUST

Spectral Analysis for Preprocessing of Data Clustering

In clustering, one common preprocessing step is to capture or predict the characteristic of target data set before the clustering algorithm is performed. Here, the spectral analysis of a data set is introduced to predict the clustering behavior before the actual data clustering. Investigating the clustering process as shown in Jain et al. (1999), an assumption is concluded that the *feature set*, and *similarity measure* embody intrinsic knowledge of the clustering domain. Data clustering algorithms are greatly dependent on the similarity matrix. Therefore, the similarity matrix can be the principal object to be considered for decoding clustering information of the data set.

Given the similarity matrix $S=(s_{ij})_{n\times n}$, we define $G(S)=<V, E, S>$ as its associated graph where V is the set of n vertices and E is the set of weighted edges. In this graph, each vertex v_i corresponds to the i-th column (or row) and the weight of each edge (v_i,v_j) corresponds to the non-diagonal entry s_{ij}. In G(S), a large weight value between two vertices represents high connectivity between them, and vice versa. In order to analyze the clustering behaviors of G(S), we employ the variant of the Laplacian T_S as the transformation of S. T_S has n eigenvalues with decreasing order $\lambda_1 \geq \lambda_2 \ldots \geq \lambda_n$, which are also called the G(S) spectrum. Two observations of the G(S) spectrum for the clustering behavior of S were found (Li et al., 2004). They indicate the relationships between the clustering behavior of S and the principal properties and structure of G(S).

1. If λ_2 is higher, there exists a better bipartition for S.
2. For the sequence

   $$\alpha_i = \frac{\lambda_i}{\lambda_2} (i \geq 2),$$

   $\exists k \geq 2$, it has $\alpha_i \rightarrow 1$ and $\alpha_i - \alpha_{i+1} > \delta$ $(1 < \delta < 1)$, then k indicates the cluster number of the data set.

They can be accounted for by spectral graph theory. The partition and connectivity of G(S) correspond in a natural way to the clustering behavior of S. Thus, through the analysis of the G(S) spectrum, we can infer details about the partition and connectivity of the graph G(S), and then obtain the clustering behavior of S. Next, we introduce the Cheeger constant, which is important in the understanding of properties and structure of G(S) (Chung, 1997). It is a typical value to measure the goodness of the optimal bipartition for a graph. Therefore, h(G) is an appropriate measure to indicate the clustering quality of the bipartition in G(S): The lower h(G) is, the better the clustering quality of the bipartition in G(S). Given the relationship between the bipartition and Cheeger constant, we have the so-called Cheeger inequality (Chung, 1997):

$$\frac{1-\lambda_2}{2} \leq h(G) \leq \sqrt{2(1-\lambda_2)}$$

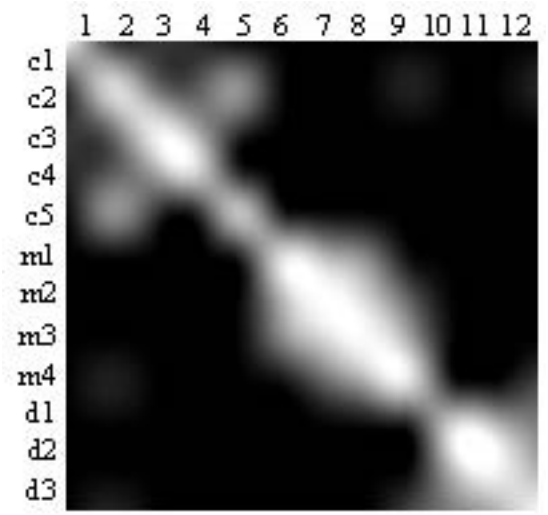
The inequality gives the bounds of h(G) by λ_2. It shows that if λ_2 is high enough to approach 1, h(G) will be very low; this indicates that there exists a good bipartition in G(S). Then we obtain Observation (1). Generally speaking, the above conclusion for G(S) is also applicable to its induced subgraphs G(S_i). Similarly, λ_2 of G(S_i) shows the clustering quality of the bipartition in G(S_i). Then we obtain Observation (2). Li et al. provided details of the theoretical and empirical results of these observations (2004).

Spectral Clustering Algorithms

Information within the spectral components is very indicative of the clustering results in the process of clustering algorithms. The intuition behind the spectral methods can be shown in the following simple example.

This example is from the example of text data when Landauer (1998) introduced LSA (p. 10). Here, we add three more passages. It uses as text passages the titles of twelve technical memoranda: five about human computer interaction (HCI), four about mathematical graph theory, and three about clustering techniques. Their topics are conceptually disjoint. We manually selected the italicized terms as the feature set and used the cosine similarity measure to compute the similarity matrix, as shown in gray scale image in *Figure 1*. The shade of each point in the image represents the value of the corresponding entry in similarity matrix. In this figure, we can see that the first, second and third diagonal blocks (white ones) correspond to the topics "HCI," "graph," and "clustering" respectively,

Figure 1. Gray scale image of the similarity matrix



while off-diagonal area shows the disjointed features of these topics.

Based on the theoretical analysis of Observation (1), the second eigenvector of the similarity matrix indicates its clustering behavior. Therefore, unlike the investigation of λ_2, we examine the eigenvector x_2, corresponding to λ_2. Certainly, the similarity needs to be transformed. After transformations, the second eigenvectors of its Laplacian and transition matrix are illustrated in *Figures 2* and *3*, respectively.

In figures, we can clearly see that the second eigenvector assigns large negative weights on the first five coordinates, large positive weights on the following four coordinates and nearly zero weights on the last three coordinates. This result is exactly identical to our class labels in *Table 1*.

This example clearly shows how the information of eigenvectors indicates the clustering behavior of a data set. In general, the eigenvectors corresponding to large eigenvalues tend to capture global clustering characteristics of a data set. Therefore, according to these intuitive observations, clustering algorithms based on spectral components can be classified into two types.

- **Recursive type:** The algorithms divide data points into two partitions based on a single eigenvector (for example, the second eigenvector) and then recursively divide each sub-partitions in the same way to find the optimum partitions.
- **Multiway type:** The algorithms use more information in multiple eigenvectors to do a direct partition of data.

Next, we will review four typical spectral clustering algorithms.

- The Shi & Malik algorithm (Shi & Malik, 2000): This algorithm was proposed as a heuristic algorithm to minimize the *normalized cut* criterion for graph partitioning in the area of image segment. Given a graph G=(V, E), the normalized cut between two sets $A \cup B = V$, $A \cap B = \varnothing$ is defined as

Figure 2. Coordinates of the second eigenvector in Laplacian

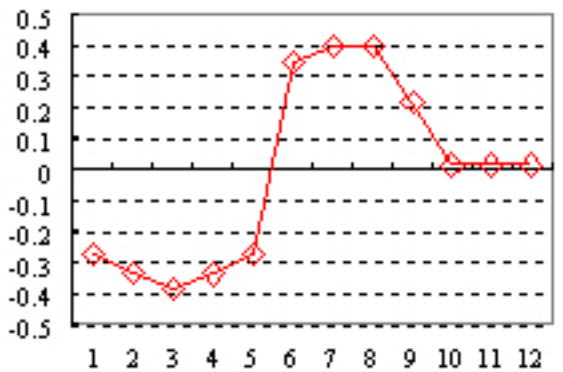


Figure 3. Coordinates of the second eigenvector in transition matrix

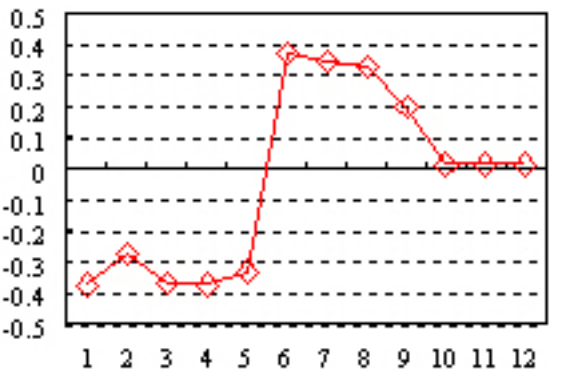


Table 1. Example of text data: Title of some technical memos

c1: *Human* machine *interface* for ABC *computer* applications
c2: A *survey* of *user* opinion of *computer system* response time
c3: The *EPS user interface* management *system*
c4: *System* and *human system* engineering testing of *EPS*
c5: Relation of *user* perceived *response time* to error measurement
m1: The generation of random, binary, ordered *trees*
m2: The intersection *graph* of paths in *trees*
m3: *Graph minors* IV: Widths of *trees* and well-quasi-ordering
m4: *Graph minors*: A *survey*
d1: An investigation of linguistic features and *clustering* algorithms for topical *document clustering*
d2: A comparison of *document clustering techniques*
d3: *Survey* of *clustering* Data Mining *Techniques*

$$\mathrm{Ncut}(A,B) = \frac{\mathrm{cut}(A,B)}{\mathrm{vol}(A)} + \frac{\mathrm{cut}(A,B)}{\mathrm{vol}(B)}$$

where the vertex set V is partitioned into two clusters A and B so that Ncut(A,B) over all two way partitions of V is minimized. This problem is proved to be NP-hard. However, Shi & Malik show that the spectral algorithm may find optimum under some special condition. Specifically, it uses the second eigenvector of the transition matrix P_S to do bipartition. This algorithm is a kind of recursive type.

- The Kannan, Vempala, & Vetta algorithm (Kannan et al., 2000): This algorithm is similar to the Shi & Malik algorithm except a key point. It uses Cheeger constant as defined in the above section to be the criterion of bipartition and the second eigenvector of the transition matrix P_S to do bipartition. Therefore, this algorithm is also one of recursive spectral clustering algorithms.
- The Meila & Shi algorithm (Meila & Shi, 2001): This algorithm is of multiway type. It first transforms the similarity matrix S to be the transition matrix P_S. Then it computes $x_1, x_2, \ldots, x_k$, the eigenvectors of P_S corresponding to the first k largest eigenvalues and generates the matrix X=[$x_1, x_2, \ldots, x_k$]. Finally it applies any non-spectral clustering algorithm to cluster rows of X as points in a k-dimensional space.
- The Ng, Jordan, & Weiss algorithm (Ng et al., 2001): This algorithm is also a kind of multiway type. It uses the variant of Laplacian matrix T_S in the transformation step. Then it computes $x_1, x_2, \ldots, x_k$, the eigenvectors of P_S corresponding to the first k largest eigenvalues and generates the matrix X=[$x_1, x_2, \ldots, x_k$]. After obtaining the matrix Y by normalizing each of X's rows to have unit length

  (i.e $y_{ij} = x_{ij} / \sqrt{\sum_j x_{ij}^2}$).

  Finally, treating each row of Y as a point in k dimensions, it clusters them by k-means clustering algorithm.

Although there are various spectral clustering algorithms, they have largely common steps and theoretical proofs: (1). All need the transformation of the similarity matrix S before the actual clustering. And they may use L_S, T_S or P_S. (2). Eigenvectors of the transformed matrix are undoubtedly used as the important data for clustering. The use of eigenvectors is also the reason why they are called spectral clustering algorithms. (3). The underlying theory of these algorithms is based on the optimization problem of graph cut. And the criterions of bipartition of the graph are introduced to prove the possible optimum solution given by eigenvectors.

FUTURE TRENDS

There is a close relationship between spectral components and a notable phenomenon frequently occurring in real-world data – power law. Distributions in much

real-life data from nature and human social behaviors, including city sizes, incomes, earthquake magnitudes, even the Internet topology, WWW and collaboration networks, which are composed of a large number of common events and a small number of rarer events, often manifest a form of regularity in which the relationship of any event to any other in the distribution scales in a simple way (Zipf, 1949; Malamud et al., 1998; Faloutsos et al., 1999; Barabási et al., 2000; Albert, 2001; Kumar et al., 2000). In essence, such distribution can be generalized as the power law, which is often represented by log-linear relations. Considering the ubiquity of power law in the real-world data, there has been a recent surge of interest in graphs whose degrees have the power-law distribution, because these graphs are often derived from real-life data. As we have discussed, spectral methods have close relationships with graphs, an intriguing problem naturally arising is: Does power-law in graphs affect the spectral methods? Actually, the eigenvalues of such graphs also follow power law, but with a little lower exponent than that of degrees (Mihail & Papadimitriou, 2002). Meanwhile, they pointed out that, "The effectiveness of several of the SVD-based algorithms requires that the underlying space has low rank, that is, a relatively small number of significant eigenvalues. Power laws on the statistics of these spaces have been observed and are quoted as evidence that the involved spaces are indeed low rank and hence spectral methods should be efficient" (Mihail & Papadimitriou, 2002, p. 8). However, this is the beginning of research on how is the effectiveness of spectral methods on real-world data. More experiments and theoretical analysis are needed.

CONCLUSION

Spectral techniques based on the analysis of the largest eigenvalues and eigenvectors have proven algorithmically successful in detecting semantics and clusters in graphs and some real-world data. In PageRank and HITS algorithms (Lawrence et al., 1999; Kleinberg, 1998), only the first eigenvector is considered and mined. In spectral clustering algorithms, the second eigenvector is proved to indicate more information about clustering behaviors in the targeted data set. However, it is worth noting that the empirical results of most spectral clustering algorithms discussed above are limited to the small or synthetic data set, not for the large scale and real-life data sets. Spectral methods shall be applied to more areas and compared with other existing methods.

REFERENCES

Azar, Y., Fiat, A., Karlin, A.R., McSherry, F., & Saia, J. (2001). Spectral analysis of data. In *The 33rd Annual ACM Symposium on Theory of Computing* (pp. 619-626). Heraklion, Crete, Greece.

Chung, F.R.K. (1997). Spectral graph theory. Number 92 in *CBMS Regional Conference Series in Mathematics*. Rhode Island: American Mathematical Society.

Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., & Harshman, R.A. (1990). Indexing by latent semantic analysis. *Journal of the American Society of Information Science, 41*(6), 391-407.

Drineas, P., Frieze, A., Kannan, R., Vempala, S., & Vinay, V. (1999). Clustering in large graphs and matrices. In *The 10th annual ACM-SIAM Symposium on Discrete Algorithms* (pp. 291-299). Baltimore, Maryland, USA.

Golub, G., & Loan, C.V. (1996). *Matrix computations*. Baltimore: The Johns Hopkins University Press.

Han, J., & Kambr, M. (2001). *Data mining concepts and techniques*. San Francisco: Morgan Kaufmann.

Jain, A.K., Murty, M.N., & Flynn P.J. (1999). Data clustering: A review. *ACM Computing Surveys, 31*(3), 264-323.

Kannan, R., Vempala, S., & Vetta, A. (2000). On clustering – good, bad and spectral. In *The 41st Annual Symposium on Foundations of Computer Science* (pp. 367-377). Redondo Beach, CA, USA.

Kleinberg, J.M. (1998). Authoritative sources in a hyperlinked environment. In *The 9th Annual ACM-SIAM Symposium Discrete Algorithms* (pp. 668-677). New York.

Landauer, T.K., Foltz, P.W., & Laham, D. (1998). Introduction to latent semantic analysis. *Discourse Processes, 25*, 259-284.

Lawrence, P., Sergey, B., Rajeev, M., & Terry W. (1999). *The PageRank citation ranking: Bringing order to*

the Web. Technical Report, Stanford Digital Library Technologies Project.

Li, W., Ng, W.-K., Ong, K.-L., & Lim, E.-P. (2004). A spectroscopy of texts for effective clustering. In *The 8th European Conference on Principles and Practice of Knowledge Discovery in Databases* (In press).

Meila, M., & Shi, J. (2001). A random walks view of spectral segmentation. In *International Workshop on AI and Statistics*.

Ng, A., Jordan, M., & Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems* (pp. 849-856).

Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22*(8), 888-905.

KEY TERMS

Adjacency Matrix: A matrix representing a graph with n vertex. It is an n-by-n array of Boolean values with the entry in row u and column v defined to be 1 if there is an edge connecting vertex u and v in the graph, and to be 0 otherwise.

Graph Invariants: Quantities to characterize the topological structure of a graph. If two graphs are topologically identical, they have identical graph invariants.

HITS Algorithm (Hypertext Induced Topic Selection): A Web search technique for ranking Web pages according to relevance to a particular search term or search phrase. Two concepts, "authority" and "hub," are proposed to characterize the importance of each Web page.

Markov Chain: A finite state machine with probabilities for each transition, that is, a probability that the next state is s_j given that the current state is s_i.

PageRank Algorithm: A Web search technique for ranking Web pages according to relevance to a particular search term or search phrase. Based on the random surfer model and Web graph, the index, PageRank, is proposed to rate the importance of each Web page to users.

Power Law Distribution: A probability distribution function, $P[X=x]\sim cx^{\pm}$, where constants $c>0$ and $\pm>0$, and $f(x)\sim g(x)$ represents that the limit of the ratios goes to 1 as x grows large.

Spectral Graph Theory: A theory on the study of the eigenvalue properties of Laplacian matrix of a graph.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 1037-1042, copyright 2005 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

Stages of Knowledge Discovery in E-Commerce Sites

Christophe Giraud-Carrier
Brigham Young University, USA

Matthew Smith
Brigham Young University, USA

INTRODUCTION

With the growth and wide availability of the Internet, most retailers have successfully added the Web to their other, more traditional distribution channels (e.g., stores, mailings). For many companies, the Web channel starts off as little more than an online catalog tied to a secure electronic point of sale. Although valuable in its own right, such use of the Web falls short of some of the unique possibilities it offers for intelligent marketing. Consider the following intrinsic differences between physical, brick-and-mortar stores, and online, Web-based stores. Physical stores are rather static and mostly customer-blind. In particular, 1) the store's layout and content are the same for all customers, 2) changes to layout and/or content are generally costly, and 3) visits are not traceable except for limited sale's data, such as what was bought, when it was bought and by what method of payment. Online stores or commercial Web sites, on the other hand, are naturally dynamic and customer-aware. Indeed, 1) layout and content can be modified easily and cheaply, 2) layout and content can be tailored to individual visitors, and 3) every visit automatically generates a rich trail of information about the customer's experience (e.g., visit duration, pages viewed, items bought if any, etc.), and possibly about the customer's persona (e.g., demographics gathered through an online questionnaire at registration time).

With such flexibility and nearly everything traceable and measurable, the Web is a marketer's dream come true. Although data-independent initiatives, such as offering social interactions (e.g., user forums) or providing virtual versions of physical stores (e.g., displays, lighting, music) (Oberbeck, 2004), can clearly enhance the user experience, the full benefit of the emerging and growing Web channel belongs to those who both gather and adequately leverage the rich information it provides.

BACKGROUND

Web mining emerged in the late 1990's as the branch of data mining concerned with the mining of data and structure on the Web. As pointed out early in the game, the transfer from traditional data mining to mining on the Web is not without significant challenges (Monticino, 1998; Spiliopoulou, 1999). Yet, many business analysts were just as quick to argue that the potential benefits far outweigh the costs (Greening, 2000; Edelstein, 2001), and researchers began developing specialized techniques and tools (Spiliopoulou & Pohle, 2001).

There have traditionally been three sub-areas of Web mining, based upon the type of data being mined: Web usage mining focuses on extracting information from users' interactions with a Web site; Web content mining focuses on extracting knowledge from the textual (and more recently, multimedia) content of Web pages; and Web structure mining focuses on discovering patterns of connection among Web pages (Kosala & Blockeel, 2000). A number of survey papers and texts have been dedicated to descriptions of various Web mining techniques and applications, as well as relevant research issues (Han & Chang, 2002; Kolari & Joshi, 2004; Scime, 2005; Liu, 2007).

In this chapter, we depart a little from this research-oriented approach. Indeed, we do not discuss the different types of data that may be mined on the Web, but rather highlight stages in the analysis of Web data, from simplest to most elaborate, so that business users may appreciate the potential of such analysis and have an implementation roadmap. At each stage, different

types of data (usage, content and structure) may be, and indeed are, used.

MAIN FOCUS

In the context of e-commerce, Web mining lends itself naturally to a staged approach, where moving from one stage to the next requires increasing sophistication, but also produces increasing return-on-investment. The first stage is limited to the analysis of the direct interaction of the user with the site; the second stage introduces behavioral information; and the third stage enables personalization of the user's experience. We examine each in turn.

Stage 1: Clickstream Analysis

The amount of data found in Web server logs is enormous and clearly evades direct human interpretation. Yet, it is rich in potential, making Web server logs the readiest source of data to analyze on a Web site (Srivastava et al., 2000; Fu et al., 2000; Moe & Fader, 2004). Web log analysis tools, such as AWStats, The Webalizer, SiteCatalyst, ClickTracks, Google Analytics, and NetTracker have been designed specifically to summarize and interpret Web server log data, allowing marketers to gain basic knowledge about e-commerce customers' activities, including unique number of visitors and hits, visit duration, visitors' paths through the site (i.e., clickstream), visitors' host and domain, search engine referrals, robot or crawler visits, visitors' browser and operating system, and search keywords used. These clickstream analysis reports may provide insight into business questions such as:

- Where do most visitors come from?
- What proportion of visitors come from a direct link or bookmark, a search engine, or a partner Web site (if any)?
- Which search engines (e.g., Google, Yahoo, MSN, etc.) and search terms are most frequently used?
- How long do visitors stay on the Web site?
- How many pages do visitors see on average?
- Which pages are most popular?
- How does Web site activity evolve throughout the day, week or month?
- From which pages do visitors commonly exit the Web site?

This information in turn helps e-retailers better understand the dynamics of customers' interactions with online offerings, and aids in such decisions as: which referrer to invest in, which pages to remove or replace, which pages to improve, etc. For instance, if clickstream analysis shows that a substantial number of visitors are accessing content several clicks deep into the Web site, then it might be valuable to make that content more accessible to visitors (e.g., maintaining and linking to "Top Sellers," "Most Wished for Items," or "Most Popular Items" pages on the home page).

Stage 2: Behavior Analysis

In general, transactional data and order information are not stored directly in standard Web server logs. Yet, both are essential in discovering patterns of buying and non-buying customers. The next stage of knowledge discovery requires linking clickstream data to order information. With adequate design, a host of new business-relevant questions may be answered when the front-end clickstream data is linked to, and mined together with, the back-end transactional data, allowing marketers to take further control of their e-commerce activity (Mobasher et al., 1996; Gomory et al., 1999; Rusmevichientong et al., 2004). The following are a few classical examples.

- What is the conversion rate (i.e., how many Web site visitors become buying customers)?
- How many would-be customers begin shopping (i.e., partially filling up their shopping cart) but drop out before proceeding to or completing check-out?
- How well did special offer *X* do (i.e., how much revenue vs. interest did it generate)?
- Who buys product *P*?
- Who are the most profitable customers?
- What is being bought by whom?

Answers to these questions help e-retailers better understand their customers' buying behavior as well as the value of their offerings. This information, in turn, leads to considerations such as what products to focus on, how to turn browsers into customers, how

to recognize customer types so as to respond to their needs/interests on-the-fly, etc.

It is important to recognize that behavior analysis cannot be an after-thought. Indeed, it must be carefully planned and reflected in the e-commerce site's design. For example, assume an e-retailer wishes to run a special offer and measure its impact. Then, a mechanism capable of capturing the supporting data must be set in place. In the online world, this may be accomplished simply and cost-effectively by adding a special ad somewhere on the Web site and tracking the visitors who click on it. Tracking is easily done using a distinguisher in the URL, yet this must be reflected in the implementation prior to launching the special offer. Once the mechanism is in place, one can easily measure the offer's success rate by computing the ratio of those who click and convert (i.e., buy the special offer) to those who only click.

Stage 3: Personalization

Every marketer's dream is to know enough about customers so as to tailor offers to each individually, in terms of both products and prices. Even when the needed knowledge is available, this is nearly impossible in a traditional store. Internet technology, on the other hand, makes it possible to adapt layout, contents and services offered "to the needs of a particular user or a set of users, taking advantage of the knowledge gained from the users' navigational behavior and individual interests, in combination with the content and the structure of the Web site" (Eirinaki & Vazirgiannis, 2003). This kind of personalization is the final stage of knowledge discovery in e-commerce (Perkowitz & Etzioni, 1999; Mobasher et al., 2000; Allen et al., 2001; Eirinaki & Vazirgiannis, 2003; Linden et al., 2003). It can lead to finely-honed marketing actions, such as:

- Show product *P* to customer *C*.
- Offer a discounted price on bundle *B* to customer *C*.
- Suggest services and products that customer *C* is likely to be interested in.
- Provide timely chat or co-browsing to most valuable customers (Beagle Research Group, 2004).

Interestingly, with dynamic Web design, one may use both information previously obtained and data provided in real-time (i.e., as the visitor interacts with the site) to tailor the exchange between the parties. Some of the largest e-businesses have had enormous success personalizing Web content. For instance, Google presents relevant advertisements based on keywords in which a visitor is interested; Amazon.com uses collaborative filtering, based on clusters of users with similar buying behavior, to recommend products to users on-the-fly; and Yahoo!'s LAUNCHcast makes recommendation to listeners by clustering music ratings elicited explicitly from visitors.

Ideally, personalization benefits both the customer and the e-retailer. Successful recommendations benefit customers by readily providing them with items most likely to be of interest, sometimes even introducing items that they were previously unaware of. In fact, not only are the most relevant products delivered but the transaction itself requires less time, thus improving overall customer satisfaction. In turn, more visitors will convert (i.e., buy what is suggested to them) and existing customers will return (i.e., build loyalty), so that the e-retailer sees increased revenue and profit at a relatively small cost.

FUTURE TRENDS

From the content mining perspective, multi-lingual and multimedia Web sites offer a number of interesting challenges to the Web mining research community. Advanced text mining, including the use of semantics (e.g., Web 2.0) as well as image processing techniques may serve as useful tools to enhance Web mining.

From the structure mining perspective, interesting opportunities exist within the idea of social networks. As people interact with one another, either explicitly or only implicitly, through one or more Web sites, analysts may begin to build social networks (Wasserman & Faust, 1994). Leveraging customer social networks presents additional opportunities for strategic personalized marketing. Identifying the group(s) to which a customer belongs is an effective method for personalizing content. Furthermore, since it appears that some customers have a disproportionate ability to shape public opinion (Dye, 2000), social networks

can be utilized to identify such individuals as well as the customers they influence. After identifying these groups and the influential people within them, ads and products can be strategically targeted to effectively create "buzz" (Rosen, 2000).

CONCLUSION

E-commerce activities have much richer data footprints than their physical counterparts, thus providing much greater opportunities for intelligent e-business. In order to be successful, knowledge discovery has to be planned for in the design phase of e-commerce applications. It can then be leveraged through a series of incremental stages, intended to bring increasing return-on-investment at each stage and to help e-retailers get closer to an optimal use of the Web channel. Indeed, the unique nature of the online world makes achievable the double objective of maximizing the customer's experience and maximizing revenues.

Great success has already been achieved, both technically and commercially, with Web mining. A thorough analysis of lessons learned and typical challenges faced, based on a number of implementations for a variety of customers, was published recently (Kohavi et al., 2004). Much can still be done, but thanks to the Internet's flexibility, e-commerce is probably the closest one can hope for to get back to Mr. Miller's corner shop's successful one-to-one marketing (Cabena et al., 1997).

REFERENCES

Allen, C., Kania, D., & Yaeckel, B. (2001). *One-to-One Web Marketing: Build a Relationship Marketing Strategy One Customer at a Time*, John Wiley & Sons, 2nd Edition.

Beagle Research Group (2004). Turning Browsers into Buyers with Value-Based Routing: Methodology Enhanced e-Commerce, Beagle Research Group White Paper. (http://www.beagleresearch.com/DownloadPDFfiles/ProficientFINAL.pdf).

Cabena, P., Hadjinian, P., Stadler, R., Verhees, J., & Zanasi, A. (1997). *Discovering Data Mining: From Concept to Implementation*, Prentice Hall.

Dye, R. (2000). The Buzz on Buzz, *Harvard Business Review*, November-December.

Edelstein, H.A. (2001). Pan For Gold In The Clickstream, *Information Week*, 12 March. (http://www.informationweek.com/828/prmining.htm).

Eirinaki, M., & Vazirgiannis, M. (2003). Web Mining for Web Personalization, *ACM Transactions on Internet Technologies*, 3(1):1-27.

Fu, X., Budzik, J., & Hammond, K.J. (2000). Mining Navigation History for Recommendation, in *Proceedings of the International Conference on Intelligent User Interfaces*, 106-112.

Gomory, S., Hoch, R., Lee, J., Podlaseck, M., & Schonberg, E. (1999). E-Commerce Intelligence: Measuring, Analyzing, and Reporting on Merchandising Effectiveness of Online Stores, Technical Report, IBM T.J. Watson Research Center.

Greening, D.R. (2000). Data Mining on the Web: There's Gold in that Mountain of Data, *New Architect Daily*, January. (http://www.webtechniques.com/archives/2000/01/greening/).

Han, J., & Chang, C.-C. (2002). Data Mining for Web Intelligence, *Computer*, November, 54-60.

Kohavi, R., Mason, L., Parekh, R., & Zheng, Z. (2004). Lessons and Challenges from Mining Retail E-Commerce Data, *Machine Learning*, 57(1-2):83-113.

Kolari, P., & Joshi, A. (2004). Web Mining: Research and Practice, *Computing in Science and Engineering*, 6(4):49-53.

Kosala, R., & Blockeel, H. (2000). Web Mining Research: A Survey, *SIGKDD Explorations*, 2(1):1-15.

Linden, G., Smith, B., & York, J. (2003). Amazon.com Recommendations: Item-to-Item Collaborative Filtering, *IEEE Internet Computing*, Industry Report, January/February.

Liu, B. (2007). *Web Data Mining: Exploring Hyperlinks, Contents and Usage Data*, Springer.

Mobasher, B., Jain, N., Han, E., and Srivastava, J. (1996). Web Mining: Pattern Discovery from World Wide Web Transactions, Technical Report TR-96050, Department of Computer Science, University of Minnesota, Minneapolis.

Mobasher, B., Cooley, R., & Srivastava, J. (2000). Automatic Personalization Based on Web Usage Mining, *Communications of the ACM*, 43(8):142-151.

Moe, W., & Fader, P. (2004). Capturing Evolving Visit Behavior in Clickstream Data, *Journal of Interactive Marketing*, 18(1):5-19.

Monticino, M. (1998). Web-analysis: Stripping Away the Hype, *Computer*, 31(12):130-132.

Oberbeck, S. (2004). Internet Shopping Is Big Business, but More Can Be Done by E-retailers. *The Salt Lake Tribune*, December.

Perkowitz, M., & Etzioni, O. (1999). Towards Adaptive Web Sites: Conceptual Framework and Case Study, *Computer Networks*, 31(11-16):1245-1258.

Rosen, E. (2000). *The Anatomy of Buzz*, New York: Doubleday.

Rusmevichientong, P., Zhu, S., & Selinger, D. (2004). Identifying Early Buyers from Purchase Data, In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 671-677.

Scime, A. (2005). *Web Mining: Applications and Techniques*, Idea Group Inc.

Spiliopoulou, M. (1999). The Laborious Way from Data Mining to Web Mining, *International Journal of Computer Systems, Science & Engineering*, 14:113-126.

Spiliopoulou, M., & Pohle, C. (2001). Data Mining for Measuring and Improving the Success of Web Sites, *Data Mining and Knowledge Discovery*, 5(1-2):85-114.

Srivastava, J., Cooley, R., Desphande, M., & Tan, P-M. (2000). Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data, *SIGKDD Explorations*, 1(2):12-23.

Wasserman, S., & Faust, K. (1994). *Social Network Analysis: Methods and Applications*, Cambridge University Press.

KEY TERMS

Clickstream Analysis: Analysis of the "footprint" left by visitors to a Web site. This footprint, generally referred to as a clickstream, is stored in the Web server log file.

Data Mining: Application of visualization, statistics and machine learning to the discovery of knowledge in databases. There is general consensus that the knowledge found through data mining should in some way be novel and actionable.

E-Commerce: Set of commercial activities (e.g., marketing, sales) conducted over the Internet.

Web Mining: Application of Data Mining techniques to Web data, generally consisting of usage data, content and structure. Web usage mining attempts to discover patterns of behavior (e.g., navigation) from user interactions with a Web site. Web content mining focuses on extracting knowledge from the actual content (e.g., text) of Web sites. Web structure mining considers links across Web sites to identify interesting graph patterns (e.g., hubs, authorities).

Web Server Log File: A file maintained by the Web server that stores information about visitors' interaction with the Web site. It consists of such data as basic identifying information (e.g., originating IP address) as well as time-stamped entries for all pages visited, from the time a visitor enters the site to the time he/she leaves it.

Statistical Data Editing

Claudio Conversano
University of Cagliari, Italy

Roberta Siciliano
University of Naples, Federico II, Italy

INTRODUCTION

Statistical Data Editing (SDE) is the process of checking and correcting data for errors. Winkler (1999) defines it the set of methods used to edit (clean-up) and impute (fill-in) missing or contradictory data. The result of SDE is data that can be used for analytic purposes.

Editing literature goes back to 60's with the contributions of Nordbotten (1965), Pritzker et al. (1965) and Freund and Hartley (1967). A first mathematical formalization of the editing process is in Naus et al. (1972), who introduce a probabilistic criterion for the identification of records (or the part of them) that failed the editing process. A solid methodology for generalized editing and imputation systems is developed in Fellegi and Holt (1976). The great break in rationalizing the process came as a direct consequence of the PC evolution in the 80's: Editing started to be performed on-line on PCs even during the interview and by the respondent in computer assisted self-interviewing (CASI) models of data collection (Bethlehem et al., 1989).

Nowadays, SDE is a research topic in academia and statistical agencies. The European Economic Commission periodically organizes a workshop on the subject concerning both scientific and managerial aspects of SDE (www.unece.org/stats).

BACKGROUND

Before the computers advent, editing was performed by large groups of persons undertaking very simple checks and detecting only a small fraction of errors. The computers evolution allowed survey designers and managers to review all records by consistently applying even sophisticated checks to detect most of the errors in the data that could not be found manually. The focus of both methodologies and applications was on the possibilities of enhancing the checks and of applying automated imputation rules to rationalize the process.

SDE Process

Statistical organizations periodically perform a SDE process. It begins with data collection. An interviewer can quickly examine the respondent answers and highlight gross errors. Whenever data collection is performed using a computer, more complex edits can be stored in it in advance and can be applied to data just before their transmission to a central database. In such cases, the core of editing activity is performed after completing data collection. Nowadays, any modern editing process is based on the a-priori specification of a set of edits, i.e., logical conditions or restrictions on data values. A given set of edits is not necessarily correct: important edits may be omitted and conceptually wrong, too restrictive or logically inconsistent edits may be included. The extent of these problems is reduced by a subject-matter expert edits specification. Problems are not eliminated, however, because many surveys involve large questionnaires and require the complex specification of hundreds of edits. As a check, a proposed set of edits is applied on test data with known errors before application on real data. Missing edits or logically inconsistent ones, however, may not be detected at this stage. Problems in the edits, if discovered during the actual editing or even after it, cause editing to start anew after their correction, leading to delays and incurring larger costs than expected. Any method or procedure which would assist in the most efficient specification of edits would therefore be welcome.

The final result of a SDE process is the production of clean data and the indication of the underlying causes of errors in the data. Usually, an editing software is able to produce reports indicating frequent errors in the data. The analysis of such reports allows to investigate the data error generation causes and to improve the results

of future surveys in terms of data quality. Elimination of sources of errors in a survey allow a data collector agency to save money.

SDE Activities

SDE concerns two aspects of data quality; (1) Data Validation: the correction of logical errors in the data; (2) Data Imputation: the imputation of correct values once errors in data have been localized. Whenever missing values appear in data, missing data treatment is part of the data imputation process to be performed in the SDE framework.

Types of editing

The different 'kinds' of editing activities are:

- **Micro Editing:** The separate examination of each single record for the assessment of the logical consistency of data, using a mathematical formalization in the automation of SDE.
- **Macro Editing:** Examination of the relationships between a given data record and the others, in order to account for the possible presence of errors. A classical example is outlier detection, i.e. the examination of the proximity between a data value and some measures of location of the distribution it belongs to. Outlier detection literature is vast and it is possible to refer to any of the classical text in the subject (for instance Barnett and Lewis, 1994). For compositional data, a common outlier detection approach is provided by the aggregate method, aimed to identify suspicious values (i.e. possible errors) in the total figures and to drill-down to their components to figure out the sources of errors. Other approaches use both data visualization tools (De Waal et al., 2000) and statistical models describing changes of data values over the time or across domains (Revilla and Rey, 2000).
- **Selective Editing:** An hybrid between micro and macro editing: the most influential among the records that need imputation are identified and their correction is made by human operators, whereas remaining records are automatically imputed by the computer. Influential records are often identified looking at the characteristics of the corresponding sample unit (e.g. large companies in an industry survey) or applying the "score variable method" (Hidiroglou and Berthelot, 1986) that accounts for the influence of each subset of observations on the estimates produced for the whole dataset.
- **Significance Editing:** A variant of selective editing introduced by Lawrence and McKenzie (2000). The influence of each record on the others is examined at the moment the record is processed and not after all records have been processed.

MAIN THRUST

Editing literature does not contain relevant suggestions. The Fellegi-Holt method is based on set theory concepts, which helps to perform efficiently several steps of the process. This method represents a milestone, since all recent contributions are aimed to improve it, particularly its computational effectiveness.

The Fellegi-Holt Method Data

Fellegi and Holt (1976) provide a solid mathematical model for SDE in which all edits reside in easily maintained tables. In conventional editing, thousands of lines of if-then-else code need to be maintained and debugged.

In the Fellegi-Holt model, a set of edits is a set of points determined by edit restraints. An edit is failed if a record intersects the set of points. Generally, discrete restraints are defined for discrete data and linear inequality restraints for continuous data. An example for continuous data is:

$$\sum_i a_{ij} x_j \le C_j, \ \forall j = 1, 2, \ldots\ldots, n$$

whereas for discrete data an edit is specified in the form $\{Age \le 15, \ marital\ status = Married\}$. The record *r* falling in the set of edit restraints fails the edit. It is intuitive one field (variable) in a record *r* must be changed for each failing edit. A major difficulty arises if fields (variables) associated with failing edits are changed: then, other edits that did not fail originally will fail.

The mathematical routines code in the Fellegi-Holt model can be easily maintained. It is possible to check

the logical validity of the system prior to the receipt of data. In one pass through the data of an edit-failing record, it is possible to fill in and change values of variables so that the record satisfies all edits.

Checking the logical validity is often referred to as determining the consistency or logical consistency of a set of edits. The Fellegi-Holt goals are:

1. Data in each record should be made able to satisfy all edits by changing the fewest possible variables.
2. Imputation rules should derive automatically from edit rules.
3. When imputation is necessary, it should maintain the joint distribution of variables.

Goal 1 refers to the *error localization*. The Fellegi-Holt method requires the generation of all the implicit and explicit edits for its solution. Explicit edits are generated by subject matter expert according to the nature of the variables, whereas implicit edits are derived from a set of explicit edits. If the implicit edit fails, then necessarily at least one of the explicit edits used in the generation of the implicit ones fail. The Fellegi-Holt method main hint is the demonstration of a theorem about the possibility of finding a set of fields to change in a record that yield a changed record satisfying all edits.

If a complete set of implicit edits can be logically derived prior to editing, then the integer programming routines that determine the minimal number of fields to change in a record are relatively fast. Generally, the derivation of all implicit edits prior to editing is difficult (Garfinkel et al., 1986). When most of the implicit edits are available, Winkler and Chen (2002) describe an efficient way to determine the approximate minimal number of fields to change.

Fellegi and Sunter (1969) show that implicit edits provide information about edits that do not originally fail but may fail as a record is changed.

Improving the Speed of the Implicit Edit Generation

Systems employing the Fellegi-Holt method have been mainly developed for categorical variables. The main problem concerns the implicit edit generation, since the computational time is a steep exponential function of the number of explicit edits. A common but not completely satisfactory solution is to split the set of explicit edits into subsets and generate implicit edits separately for each subset. Editing systems employing the Fellegi-Holt method for categorical variables usually works by splitting the set of explicit edits. These systems are used in Italy (Barcaroli et al., 1997), Spain and Canada. Garfinkel et al. (1986) provide an algorithm (implemented by the U.S. Census Bureau) for reducing the amount of computation required for implicit edit generation by identifying in advance, for each candidate set of contributing edits and generating field, those subsets (prime covers) that have a possibility of producing the maximal number of new edits. Prime covers are groups of edits which do not have any subsets with the same properties. For each set of contributing edits there may exist more than one prime covers. Anyway, these methods often fail producing all implicit edits when dealing with survey questionnaire presenting complicated skip patterns.

Error Localization Using an Incomplete Set of Edits

Some approaches do not generate the complete set of edits but attempt to perform error localization on the basis of an incomplete set of edits.

Winkler and Chen (2002) provide an heuristic to perform iteratively error localization when some implicit edits are missing. In practice, starting from a subset of implicit edits it is possible to detect new ones from the explicit edits failed by a given data record. The error localization process stops as soon as a certain data record does not fail any explicit edit.

As for edits involving numerical variables, great difficulties arises for the generation of implicit edits. Efforts have concentrated only on linear and ratio edits applying the Chernickova (1964) algorithms. Some slight modifications of these algorithms have been implemented by the major statistical agencies (Canada, The Netherlands, USA).

Other approaches are based on statistical tools, in particular tree-based models (Conversano and Cappelli, 2002) and nearest-neighbors methods (Bankier et al., 2000).

Careful Design and Evaluation of the Set of Query Edits

The design of the entire set of edits is particularly important to get an acceptable cost/benefit outcome of editing. Edits have to be coordinated for related items and adapted to the date to be edited. Probable measures for improving current processes are relaxing bounds by replacing subjectively set limits by bounds based on statistics from the data to be edited, and removing edits which produce unnecessary flags. It should be noted there is a substantial dependence between edits and errors for related items. Furthermore, edits have to be targeted on the specific error types of the survey, not on possible errors.

SDE for Exceptionally Large Files

A model-based SDE procedure is proposed in Petrakos et al. (2004) within the Knowledge Discovery from Databases (KDD) and Data Mining framework. It uses recursive partitioning algorithms to perform SDE. This approach results in a fully automated procedure inspired by the *Total Quality Management* (TQM) principles ("*Plan, Do, Check, Act*") of the well-known "Deming Cycle" named *TreeVal*, that can be used for the derivation of edits not requiring, at least initially, the help of subject matter experts. An application on real data provided by the Official Statistical Institute of Portugal documents about its effectiveness.

As for periodic surveys, a *Survey Database* stores data deriving from previous similar surveys. It is made up of "clean data", namely data that were validated in the past. Instead, the *Incoming Data* contains cases that must be validated before being included into the Survey Database.

To simplify the SDE process, the validation procedure is not applied on the whole survey data *tout-court*, but on subsamples of cases/variables of the Survey Database which are selected according to a specific data selection and validation planning (*Plan*). These subsamples concur to define the *Pilot Dataset*, used to derive edits through the FAST recursive partitioning algorithm (*Do*) introduced by Mola and Siciliano (1997). The corresponding subsamples of the Incoming Data are selected (*Validation Sample*) and validated using edits derived in the previous step (*Check*). The clean *Validated Data* are stored in the Survey Database and used to edit subsequent subsamples (*Act*).

Another strictly related SDE field is Data Fusion (DF, Saporta, 2002). It consists in merging information collected from different sample surveys and is based on two samples: the *donor file* is a complete data matrix (X) related to a first survey; the *receptor file* (Y) is similar to X except for a certain number of missing variables. Missingness is due to lack of an entire part of the second survey. The goal is to complete Y beginning from the knowledge acquired from X. Consequently, DF is a particular case of data imputation since a group of instances is missing because it was not originally collected. Thus, two different and independent databases have to be merged. Several approaches to DF exist, some based on factor analysis (Aluja-Banet et al., 1997), others on tree-based methods (D'Ambrosio et al., 2007) .

FUTURE TRENDS

As a matter of fact, editing is a part of the data total quality improvement process, not the whole data quality process. In fact, editing alone cannot detect all errors, and definitively not correct all mistakes committed during the survey design, data collection and processing.

For future application of SDE in Business Intelligence the following editing rules, to be performed in priority order, have to be specified:

1. Identify and collect data on problem areas, and error causes in data collection and processing, producing the basics for the (future) improvement of the survey vehicle.
2. Provide information about the quality of data.
3. Identify and handle concrete important errors and outliers in individual data.

Besides its basic role to eliminate fatal errors in data, SDE should highlight, not conceal, serious problems in the survey vehicle. The focus should be on the cause of an error, not on the particular error *per se*.

CONCLUSION

Since data need to be free of errors before being analyzed in a statistical way, SDE appears essential wherever data are collected. But SDE is not a standalone activity, since it should be integrated within data

collection, processing and estimation. Besides the goals of data correction and imputation, a main task of SDE is to provide a basis for designing measures to prevent errors.

Focusing on the recent uses of SDE, it emerges that the paradigm "the more (and tighter) checks, the better the data quality" is not always valid, since it does not exist an editing method that clearly outperform the others. When performing SDE, the entire set of the query edits should be designed meticulously, be focused on errors influencing the estimates, and be targeted on existing error types recognizable by edits. The effects of edits should be continuously evaluated by the analysis of performance measures and other diagnostics the process can be designed to produce.

REFERENCES

Aluja-Banet, T., Morineau, A., & Rius, R. (1997). La greffe de fichiers et ses conditions d'application. Méthode et exemple. In: Brossier G., Dussaix A.M. (eds), *Enquêtes et sondages*, Dunod, Paris, 94-102.

Bankier, M., Lachance, M., & Poirier, P. (2000). 2001 Canadian Census minimum change donor imputation methodology. Working paper n. 17, UN/ECE Work Session on Statistical Data Editing, http://amrads.jrc.cec.eu.int/k-base/papers.

Barcaroli, G., & Venturi, M. (1997). DAISY (Design, Analysis and Imputation System): structure, methodology and first applications. In Kovar; J., Granquist, L. (eds.), *Statistical Data Editing*, vol. 2, U.N. Economic Commission for Europe, 40-51.

Barnett, V., & Lewis, T. (1994). *Outliers in statistical data*, New York, Wiley.

Bethlehem, J.G., Hundepool, A.J., Schuerhoff, M.H., & Vermeulen, L.F.M. (1989). *BLAISE 2.0: an introduction*, Voorburg, the Netherlands: Central Bureau of Statistics.

Chernickova, N.V. (1964). Algorithm for finding a general formula for the non-negative solutions of a system of linear inequalities, *USSR Computational Mathematics and Mathematical Physics*, 4, 151-158.

Conversano, C., & Cappelli, C. (2002). Missing data incremental imputation through tree-based methods, in Härdle, W., Ronz, B. (eds.), *COMPSTAT 2002" Proceedings in Computational Statistics*, Physica-Verlag, Heidelberg, 455- 460.

D'Ambrosio, A., Aria M., & Siciliano, R. (2007). Robust tree-based incremental imputation method for data fusion. *Advances in Intelligent Data Analysis VII,*, Berthold M.R, Shawe-Taylor J., Lavrac N (eds.) Springer, 174-183.

De Waal, T. (2000). A brief overview of imputation methods applied at Statistics Netherlands, *Netherlands Official Statistics*, 3, 23-27.

Fellegi, I.P., & Holt, D. (1976). A systematic approach to automatic edit and imputation, *Journal of the American Statistical Association*, 71, 17-35.

Fellegi, I.P., & Sunter, A.B. (1969). A theory for record linkage, *Journal of the American Statistical Association*, 64, 1183-1210.

Freund, R.J., & Hartley, H.O. (1967). A procedure for automatic data editing, *Journal of the American Statistical Association*, 62, 341-352.

Garfinkel, R.S., Kunnathur, A.S., & Liepins, G.E. (1986). Optimal imputation for erroneous data, *Operations Research*, 34, 744-751.

Hidiroglou, M.A., & Berthelot, J.M. (1986). Statistical editing and imputation for periodic business surveys, *Survey Methodology*, 12, 73-84.

Lawrence, D., & McKenzie, R. (2000). The general application of significance editing, *Journal of Official Statistics*, 16, 943-950.

Mola, F., & Siciliano, R. (1997) A fast splitting algorithm for classification trees, *Statistics and Computing*, 7, 209–216.

Naus, J.I., Johnson, T.G., & Montalvo, R. (1972). A probabilistic model for identifying errors in data editing, *Journal of the American Statistical Association*, 67, 943-950.

Nordbotten, S. (1965). The efficiency of automatic detection and correction of errors in individual observations as compared with other means for improving the quality of statistics, *Proceedings of the 35-th Session of the International Statistical Institute*, Belgrade, 417-441.

Petrakos, G., Conversano, C., Farmakis, G., Mola, F., Siciliano, R., & Stavropoulos, P. (2004). New ways of specifying data edits, *Journal of the Royal Statistical Society*, A, 167, 249-264.

Pritzker, L., Ogus, J., & Hansen, M.H. (1965). Computer editing methods: some applications and results, *Proceedings of the 35-th Session of the International Statistical Institute*, Belgrade, 442-465.

Revilla, P., & Rey, P. (2000). Analysis and quality control for ARMA modelling. Working paper n. 19, UN/ECE Work Session on Statistical Data Editing, http://amrads.jrc.cec.eu.int/k-base/papers.

Saporta, G., (2002). Data fusion and data grafting, *Computational Statistics and Data Analysis*, 38, 465-473.

Winkler, W.E. (1999). State of Statistical Data Editing and current research problems. Working paper n. 29, UN/ECE Work Session on Statistical Data Editing, http://amrads.jrc.cec.eu.int/k-base/papers.

Winkler, W. E., & Chen, B. C. (2002). Extending the Fellegi–Holt model of statistical data editing. Statistical Research Report 2002/02, US Bureau of the Census,Washington DC, www.census.gov/srd/www/byyear.html.

KEY TERMS

Data Checking: Verification of the correctness conditions of data, also including the specification of the type of the error or condition not met and the qualification of the data and its division into the "error free" and "erroneous data". Data checking may be aimed at detecting error-free data or at detecting erroneous data.

Data Editing: Detection and correction of errors (logical inconsistencies) in data.

Data Fusion: Combining or merging data coming from different sources, usually different sample surveys.

Data Imputation: Substitution of estimated values for missing or inconsistent data items (fields). Substituted values are intended to create a data record that does not fail edits.

Data Validation: Verification of whether the value of a data item comes from the given (finite or infinite) set of acceptable values.

Editing Procedure: Detection and handling of errors in data, usually in three phases: definition of a consistent system of requirements, their verification on given data, and elimination or substitution of data which is in contradiction with the defined requirements.

Explicit Edit: An edit explicitly written by a subject matter specialist.

Error Localization: (Automatic) identification of the fields to impute in an edit-failing record. Usually, an optimization algorithm determines the minimal set of fields to impute, so that the final (corrected) record will not fail edits.

Implicit Edit: An unstated edit logically derived from explicit edits previously specified by a subject matter specialist.

Logical Consistency: Verification of whether a given logical condition is met. It is usually employed to check qualitative data.

Statistical Metadata Modeling and Transformations

Maria Vardaki
University of Athens, Greece

INTRODUCTION

The term metadata is frequently used in many different sciences. Statistical metadata generally used to denote "*every piece of information required by a data user to properly understand and use statistical data*." Modern statistical information systems (SIS) use metadata in relational or complex object-oriented metadata models, making an extensive and active usage of metadata. Early phases of many software development projects emphasize the design of a conceptual data/metadata model. Such a design can be detailed into a logical data/metadata model. In later stages, this model may be translated into physical data/metadata model.

Organisations aspects, user requirements and constraints created by existing data warehouse architecture lead to a conceptual architecture for metadata management, based on a common, semantically rich, object-oriented data/metadata model, integrating the main steps of data processing and covering all aspects of data warehousing (Pool et al, 2002).

In this paper we examine data/metadata modeling according to the techniques and paradigms used for metadata schemas development. However, only the integration of a model into a SIS is not sufficient for automatic manipulation of related datasets and quality assurance, if not accompanied by certain operators/ transformations. Two types of transformations can be considered: (i) the ones used to alleviate breaks in the time series and (ii) a set of model-integrated operators for automating data/metadata management and minimizing human errors. This latter category is extensively discussed.

Finally, we illustrate the applicability of our scientific framework in the area of Biomedical statistics.

BACKGROUND

Metadata and metainformation are two terms widely used interchangeably in various sciences and contexts. Until recently, metainformation was usually held as table footnotes. This was mainly due to the fact that the data producer and/or consumer had underestimated the importance of this kind of information.

When metadata integration in a pre-arranged format became evident, the use of metadata templates was proposed. This was the first attempt to capture metadata in a structured way. This approach was soon adopted since it reduced chances of ambiguous metadata as each field of the templates was well documented. However, they still had limited semantic power, as they cannot express the semantic links between the various pieces of metainformation.

To further increase the benefits of using metadata, attempts have been made to establish ways of automating the processing of statistical data. The main idea behind this task is to translate the meaning of data in a computer-understandable form. A way of achieving this goal is by using large, semantically rich, statistical data/metadata models like the ones developed in Papageorgiou et al (2001, 2002). However, in order to minimize compatibility problems between dispersed systems, the need that emerges is to build an integrated metadata model to manage data in all stages of information processing. The quantifiable benefits that have been proven through the integration of data mining with current information systems can be greatly increased, if such an integrated model is implemented. This is reinforced by the fact that both relational and OLAP technologies have tremendous capabilities for navigating massive data warehouses; nevertheless, brute force navigation of data is not enough. Such an integrated model was developed in Vardaki & Papageorgiou (2004), and it was demonstrated that an abstract, generally applied model, keeping information about storage and location of information as well as data processing steps, is essential for data mining requirements. Other related existing work focuses either mainly on OLAP databases (Pourabbas and Shoshani, 2006) or on semantically rich data models used mainly for data capturing purposes. In these cases, the authors focus their attentions on data

manipulations and maximization of the performance of data aggregations.

A number of modeling techniques and technologies have been discussed in literature and considered for the implementation of various research projects. Mainly in the case of medical statistics the Entity-Attribute-Value (EAV) (also known as "Object-Attribute-Value Model" and "Open Schema") database modeling technique has been extensively used (Dinu and Nadkarni, 2006). Also, Papageorgiou et al (2001) developed an Object Oriented metadata model for the series of processes followed in a Statistical Institute. Finally, the XML-based solution has been extensively used.

However, although a metadata model is an important step towards automation of data/metadata management and processing, the definition of a set of transformations/ operators is essential for harmonization purposes and for the automatic manipulations of statistical information. Two types of transformations can be considered: (i) the methodological transformations which are used when there are inconsistencies between practices of data collection, compilation and dissemination of statistics and (ii) operators that permit specific manipulations of the underlying data and their related metadata stored in the databases.

Regarding the methodological transformations, these are used to alleviate breaks in time series. When data collected in a specific time period are not fully comparable with the data of the following years we say that we have a break in time series. Breaks frequently occur in time series and involve changes in standards and methods that affect data comparability over time since they make data before and after the change not fully comparable. Information about breaks in time series is a quite important piece of statistical metadata because of the adverse effects they can have to statistical inference based on fragmented data. A number of transformations can be applied to minimize the effect of such incompatibility.

In case of simultaneous manipulation of both data and metadata, this is achieved by introducing a set of tools (transformations) to assist the data producer and user in manipulating both data and metadata simultaneously. Sets of such transformations have been discussed by Papageorgiou et al (2002) and Vardaki and Papageorgiou (2006).

MAIN THRUST

This chapter aims in discussing metadata modeling and related techniques and also introduce a set of underlying transformations. More specifically, topics that are covered include essential considerations in statistical metadata modeling development, modeling techniques and paradigms. Furthermore, a set of transformations is proposed for automation of data/metadata processing in a Statistical Information System.

The applicability of our scientific framework is further discussed in a case study in the area of medical statistics describing how the metadata model and the proposed transformations can allow for simultaneous handling of datasets collected during dispersed similar clinical trials performed by different medical centers.

Statistical Metadata Modeling

The design of a data/metadata model is a crucial task for further processing. If the model is undersized, it will be incapable of holding important metadata, thus, leading to problems due to missing metainformation. On the other hand, if it is oversized, it will keep information that is captured, rarely used and never updated, thus leading to severe waste of resources. Obviously, an oversized model cannot be easily implemented or used by the Institute's personnel. However, it is also difficult to predict the needs of data consumers, since the amount of required metainformation depends on the application under consideration. A step-by-step model development can serve the purpose.

We briefly consider three stages of data/metadata modeling:

- The conceptual metadata model (schema) consisting of entity classes (or objects), attributes and their relationships.
- The semantic data model describing the semantics.
- The operators/transformations abstract schema

Of course the database structure and coding as well as the data/metadata storage selection should be also represented.

Apart from choosing what metainformation is worth capturing, there is an additional difficulty in choosing the most appropriate modeling technique. Discussions

and controversial arguments have been registered in literature over the Entity-Relational (E-R) and the Object-Oriented (O-O) paradigms.

Finally, although not related to statistical modeling, we should refer to the Common Warehouse Metamodel (CWM), a specification for modeling metadata for relational, non-relational, multi-dimensional, and most other objects found in a data warehousing environment. The specification is released and owned by the Object Management Group (OMG, 2007).

Modeling Techniques and Languages

It is recommended that an integrated, semantically rich, platform-independent, statistical metadata model, should be designed to cover the major stages of the statistical information processing (data collection and analysis including harmonization, processing of data and metadata and dissemination/output phases), which can minimize complexity of data warehousing environments and compatibility problems between distributed databases and Information Systems.

Enhanced Entity-Relationship (E-R) models were developed some years ago which, in turn, proved that they lacked inheritance relationships. They can represent the entity-attribute dependency adequately; however, they lack flexibility if we need to add new metadata to reflect an additional process.

Then, the Object-Oriented (O-O) paradigm (Papazoglou et.al, 2000) has started to be used, where the statistical metadata model was described and designed in Uniform Modeling Language (UML) for a better representation of 2-level inter-dependencies (class-attribute). Unified Modeling Language (UML) is an industry standard used in modeling business concepts when building software systems using object-oriented technology. It is developed by Object Management Group (OMG, 2007).

On the other hand, Extensible Markup Language (XML) is designed especially for Web documents. It allows designers to create their own customized tags, enabling the definition, transmission, validation, and interpretation of data between applications and between organizations.

The main benefits of a metadata model designed in UML are the flexibility and that the same conceptual model may be used to generate different XML Schemas and XMI for the data level as well as other representations. Even if a new technological infrastructure is adopted, the initial conceptual UML model remains the same, or can be easily updated.

Transformations

Methodological Transformations

The *methodological transformations* include all the possible transformations that are used to enhance the data homogeneity by correcting errors that were introduced due to methodological inconsistencies and restore their effects in breaks in time series. In order to evaluate fragmentation of data in both time and space, several coefficients of fragmentation in both space and time can be developed (see also Papageorgiou et al, 2002).

Operators for Automatic Manipulation of Data/Metadata

A statistical metadata model should keep information about the series of processes that have been applied on the data of a survey (since the survey is a central characteristic in statistics). This can be achieved with the integration of several operators/ transformations. All the operators have the closure property, meaning that the application of an operator on a Survey/Dataset/Table always produces a new Survey/Dataset/Table.

The importance of operators is that they allow for simultaneous manipulation of both data and metadata. Consequently, they severely reduce the possibility of data and metadata mismatches. In addition, operations are necessary for building SISs that support metadata-guided statistical processing. In this case, the user simply describes the statistical Table of interest and the system automatically finds a series of operators that can be applied onto existing tables in order to produce the requested one.

An indicative set of transformations can be briefly described as follows (see also Vardaki & Papageorgiou, 2008):

- **Selection:** It closely resembles the one from the relational algebra. The result of applying this operator is a new table holding only a subset of the initial data satisfying the selection criterion.
- **Projection:** It also closely resembles the one from the relational algebra. The result of applying this operator is a new table holding only a subset of Variables of the initial data.

- **Reclassification:** The reclassification operator converts the values of a Variable in another classification using a different grouping criterion. However, the results of a reclassification may lead to tables with missing values, or with inaccurate values, especially when the existing values' classification is not fully convertible into the requested one.
- **Join:** The join operator resembles the join operator of the relational algebra. It is applied on two tables having one or more common variables (joined Variables). The result is a new table having all the Variables of both tables (obviously, the common Variables included once.).
- **Algebraic operators:** This is a general operator used to denote simple mathematical operations (e.g. additions, multiplications, etc.) that are frequently applied into a Table. The algebraic operator is the only one, which requires additional documentation and semantic metadata to be specified by the user, during its application. The reason is that there is no automated way for SISs to understand the semantic meaning of such an operator.

Case Study: Applicability in Biomedical Statistics

An area where metadata and metadata modeling is currently thriving is the biomedical field and especially the case of clinical and pre-clinical studies. We can find extensive discussions on the Entity-Attribute-Value (EAV) modeling regarding knowledge representation for complex heterogeneous biomedical databases (Dinu and Nadkarni, 2006).

Towards the approach of the O-O paradigm and UML representation, SCENPRO (http://www.scenpro.com/) has developed a disease-oriented (and specifically cancer-oriented) UML Object Model. The National Cancer Institute (NCI) has proposed the Cancer Biomedical Informatics Grid (caBIG) UML model and CaCORE is a related interoperability infrastructure (Komatsoulis et.al, 2007). In addition, the Clinical Data Interchange Standards Consortium (CDISC) (http://www.cdisc.org) developed a XML-based metadata model defined to guide sponsors in the preparation of data that is to be submitted to the Food and Drug Administration (FDA), thus supporting standard interchange between medical and biopharmaceutical data (Deshpande et al, 2002).

Both NCI and CDISC collaborated with Health Level Seven (HL7) and FDA experts in the promising project, BRIDG (Weng et al, 2007), for developing a UML model aiming to bridge standards and also the gap between clinical research and healthcare.

Our case study aims in demonstrating conceptually how a flexible metadata model, integrated into Clinical Study Data Management Systems (CSDMS) together with a set of proposed transformations, allows for automatic, simultaneous handling of different clinical trials performed by several centers or in various time periods.

Consider two similar medical research studies performed in outlying regions (therefore the participating populations – and the samples taken – are disjoint) and the two resulting datasets, having a number of equivalent variables (resulting from the related similar questions of the questionnaire given to the study subjects). Our conceptual approach allows only for a simplified demonstration of the transformations application on the two studies' datasets as follows:

Initially, we identify the equivalent variables in the two tabulated datasets.

Then, by applying the projection transformation to both tabulated datasets, we remove the non-equivalent variables (represented by columns in a table) from each dataset, leaving only the equivalent sets of variables. In case two equivalent variables are measured with different classifications we apply the reclassification operator to convert the measurement unit of one of them into the measurement unit of its equivalent. For example, in clinical trials performed in the United Kingdom the factor "body weight" of a patient is usually measured in pounds, whereas in Germany they measure it in kgs. If we want to combine similar trials from these two countries using kgs, we will have to apply a reclassification from pound to Kgs using the relation 1 pound = 0.4536 kgs approximately.

After that, we can derive a new survey (table) with data obtained by merging the two sets of data with equivalent variables (columns). The new survey (table) will have the Union of the two studies' study populations and the same columns (same number and description of equivalent set of variables).

We can then perform any other transformations we consider necessary like selection of specific rows (each row representing a study subject) according to

eligibility criteria, addition of variables, or join of the two studies (under certain pre-conditions), etc., or apply any algebraic transformations we want.

FUTURE TRENDS

Future plans include the potential of metadata to improve data quality as a consequence of transformations handling, as well as, on integrating the proposed model with relevant data warehousing and OLAP technologies. Therefore, the automation of statistical processing calls for the derivation of some quality metrics that will subsequently be used during the plan optimization and selection stage.

Other future research can concentrate on how to integrate metadata that can not be extracted from Data Warehousing components but resides in various other sources. Also, the harmonisation of various metadata terminologies as well as semantics integration and mappings is very promising. Similarly, the implementation of an integrated common shared metamodel is essential in order to bridge the gap between syntax and semantics of metadata representation in various software components of a Data Warehousing system.

This will lead to a possible unified adoption of standards used for data/metadata exchange, since nearly each house uses a different metadata model that most of the times is compatible only with software provided by the same software vendor.

Regarding non-classical data analysis methods, like for example the symbolic analysis, the the metadata model metainformation for the classical and the symbolic data, keeping also the processing history of original and symbolic data manipulations and the application of certain statistical methods or/and the visualization of the final results (Papageorgiou and Vardaki, 2008).

Regarding the biomedical sector, a metadata model describing the entire process of a medical research seems very promising. A number of equivalencies and transformations should be integrated for automatic combination of dispersed medical studies' data.

CONCLUSION

Structured metadata are indispensable for the understanding of statistical results. The development of a flexible, platform-independent metadata model, integrated into a SIS, is essential for the improvement of the information system and is widely considered as promising for improving effectiveness and efficiency of managing data warehouse environments. The support of automated, metadata-guided processing is vital for the next generation of statistical web sites as well as for asserting data quality.

We discussed current modeling techniques and paradigms and a number of transformations which, if properly used, can minimize the amount of data exposed to network, thus reducing problems concerning confidentiality and security of data. The reason is that computers will be able to assist the user in manipulating and processing both data and metadata. Thus, human intervention and the possibility of common human errors will be reduced. Therefore, an enhancement of services' quality and a minimum need for highly trained personnel are expected.

In our case study in the area of medical statistics we illustrated that the automatic, simultaneous manipulation of dispersed data is achieved with the use of the introduced transformations.

REFERENCES

Deshpande A, Brandt C and Nadkarni P. (2002). Metadata-driven ad hoc query of patient data: meeting the needs of clinical studies. *J.Am.Med.Info. Assoc. 9*, 369–382.

Dinu V. and Nadkarni P. (2006). Guidelines for the effective use of entity–attribute–value modeling for biomedical databases. *Int. J. Med. Inform.* [Epub ahead of print]

Komatsoulis G, Warzela D, Hartela F, Shanbhaga K, Chilukuric R, Fragoso G, Coronado S, Reeves D., Hadfield J, Ludet C and Covitz P. (2007). caCORE version 3: Implementation of a model driven, service-oriented architecture for semantic interoperability, *J Biomed Inform.* [Epub ahead of print].

OMG. *Unified language specification*, Object Management Group (OMG) Inc. The Internet: <http://www.omg.org>.

Papageorgiou, H., Pentaris, F., Theodorou E., Vardaki M. & Petrakos M. (2001). A statistical metadata model for simultaneous manipulation of data and metadata,

Journal of Intelligent Information Systems (JIIS), 17(2/3), 169-192.

Papageorgiou, H. and Vardaki M. (2008). . A statistical metadata model for symbolic objects. *Symbolic data analysis and the SODAS software*. John Wiley & Sons (forthcoming)

Papageorgiou, H., Vardaki M., Theodorou E. & Pentaris, F. (2002). The use of statistical metadata modelling and related transformations to assess the quality of statistical reports, *Joint UNECE/Eurostat Seminar on Integrated Statistical Information Systems and Related Matters* (ISIS 2002), Geneva, Switzerland, The Internet <www.unece.org/stats/documents/ces/sem.47/24.e.pdf>.

Papazoglou M.P., Spaccapietra S. & Tari Z. (2000). *Advances in object-oriented data modeling.* Cambridge, MA: The MIT Press. ISBN 0-262-16189-3.

Pourabbas E. and Shoshani A. (2006). The Composite OLAP-object data model: Removing an unnecessary barrier. *Eighteenth International Conference on Scientific and Statistical Database Management* (SSDBM), (pp. 291-300). Vienna, Austria: IEEE Computer Society.

Shoshani A. (2003). Multidimensionality in statistical, OLAP, and scientific databases. In Maurizio Rafanelli (ed.) *Multidimensional databases: Problems and solutions*, (pp. 46-68). Hershey, PA: IGI Global Publishing.

Vardaki M. (2005) Statistical metadata in data processing and interchange. In J. Wang, (Ed.), *Encyclopedia of Data Warehousing and Mining*. vol. 2, (pp. 1048-530. Herhsey, PA: Information Science Reference.

Vardaki M. and Papageorgiou H. (2008). Statistical data and metadata quality assessment. In G.D. Garson & M. Khosrow-Pour (Eds.), *Handbook of Research on Public Information Technology*, Hershey, PA: Information Science Reference. (forthcoming)

Vardaki M. & Papageorgiou H. (2004). An integrated metadata model for statistical data collection and processing. In *Sixteenth International Conference on Scientific and Statistical Database Management* (SSDBM), (pp. 363-372), Santorini, Greece. IEEE Computer Society.

Weng C, Gennari J, Fridsma D. (2007). User-centered semantic harmonization: A case study. *J Biomed Inform., 40,* 353–364.

KEY TERMS

Common Warehouse Metamodel (CWM): A specification for modeling metadata for relational, non-relational, multi-dimensional, and most other objects found in a data warehousing environment. The specification is released and owned by the Object Management Group.

DataInterchange: The process of sending and receiving data in such a way that the information content or meaning assigned to the data is not altered during the transmission.

Data Processing: The operation performed on data in order to derive new information according to a given set of rules.

Metadata Modeling: The analysis, construction and development of the frames, rules, constraints, models and theories applicable and useful for the modeling in a predefined class of problems.

Meta-Object Facility (MOF): An Object Management Group (OMG) standard for Model Driven Engineering. The official reference page may be found at OMG's MetaObject Facility. MOF originated in the need of a Metamodeling architecture to define the UML. MOF is designed as a four-layered architecture. It provides a meta-meta model at the top layer.

Platform- Independent Model: A platform-independent model or PIM is a model of a software or business system that is independent of the specific technological platform used to implement it.

Statistical Information System: It is the information system oriented towards the collection, storage, transformation and distribution of statistical information.

Statistical Metadata: They are data about statistical data. Metadata provide information on data and about processes of producing and using data. Metadata describe statistical data and - to some extent - processes and tools involved in the production and usage of statistical data.

Transformations: A set of operators to assist the user in manipulating both data and metadata simultaneously

XML: Extensible Markup Language (XML) is a general-purpose markup language. It is classified as an extensible language because it allows its users to define their own tags. Its primary purpose is to facilitate the sharing of structured data across different information systems, particularly via the Internet.

Statistical Models for Operational Risk

Concetto Elvio Bonafede
University of Pavia, Italy

INTRODUCTION

A statistical model is a possible representation (not necessarily complex) of a situation of the real world. Models are useful to give a good knowledge of the principal elements of the examined situation and so to make previsions or to control such a situation.

In the banking sector, models, techniques and regulations have been developed for evaluating Market and Credit risks, for linking together risks, capital and profit opportunity. The regulations and vigilance standards on the capital have been developed from the Basel Committee founded at the end of 1974 by the G10.

The standards for the capital's measurement system were defined in 1988 with the "Capital Accord" (BIS, 1988); nowadays, it is supported from over 150 countries around the world. In January 2001 the Basel Committee published the document "The New Basel Capital Accord" (BIS, 2001), which is a consultative document to define the new regulation for the bank capital requirement. Such a document has been revisited many times (see BIS, 2005).

With the new accord there is the necessity of appraising and managing, beyond the financial risks, also the category of the operational risks (OR) already responsible of losses and bankruptcies (Cruz (Ed.), 2004; Alexander (Ed.), 2003; Cruz, 2002).

BACKGROUND

The operational risk (OR), according to the new Basel accord, is due to detrimental events caused by the inadequacy or the failure of internal processes and systems, human errors and external events, for instance natural calamity (BIS, 2005).

The evaluation of a suitable risk profile is important, because banks with the same levels of market and credit risk can have a different OR profile. The operational risk, in fact, is an intrinsic characteristic of the bank, of the performed activities and of the place in which the institution is located (Cruz (Ed.), 2004).

Due to the peculiarity of OR, the difficulties that are peculiar to its modelling are the following:

1. The OR set is heterogeneous and strongly dependent of the context where it is valued.
2. Some events, which are referable to the OR, produce damages that are hardly evaluable.
3. Some OR events are very rare. Probably the single bank has never faced such events, and in this case the institution needs also external data.
4. For some events the past history is not a good indication of the future.
5. Lack of reliable historical data.
6. Associated problems with events' estimate that have high frequency and low impact (HFLI) and vice versa with low frequency and high impact (LFHI).

Besides, the greatest problems arise from the organization of the database (DB) for the construction and validation of the models, (Cruz (Ed.), 2004; Alexander (Ed.), 2003).

The Basel Committee with the new accord recommends the use of three methods for the valuation of the *value at risk* (VaR) characterized by increasing complexity: base (*BIA*), standard (*STA*) and advanced (AMA), (BIS, 2005; Cornalba & Giudici 2004).

Such approaches are subjected to criticisms due to the difficulties to evaluate operational risk and for the way in which they influence the capital (necessary to cover OR) in function of the institution amplitude (Cruz (Ed.), 2004; Alexander (Ed.), 2003).

MAIN FOCUS

The AMA approach is more complex, but it makes the calculation of the *value at risk* (VaR) more sensitive to the risk profile and generally smaller than the approach calculated with *BIA* and *STA*. Every bank can use its advanced internal model if it satisfies the qualitative

and quantitative standards defined by the new accord (BIS, 2005; BIS, 2003).

The AMA methods are bottom-up type and this because the VaR calculation is achieved considering the losses obtained by dividing the bank's activities in eight business lines (BL) and seven event types (ET or risk category). In this manner there will be at least 56 different kind of losses, one for each intersection BL/ET.

The models for the AMA approaches are divided in two principal classes, quantitative and qualitative models. The actuarial and analytical models represent the former; the latter are constituted by Scorecard Approach (SA). Bayesian methods are placed between the two categories, (Fanoni, Giudici & Muratori, 2005; Giudici & Bilotta, 2004; Cornalba & Giudici, 2004; Cruz (Ed.), 2004; Alexander (Ed.), 2003; Cruz, 2002).

To quantify the risk is necessary to know the statistical distribution of the number of risky events (frequency) and the statistical distribution of theirs consequences (severity or impact).

All the models have to be validated with *scenario analysis* and *backtesting*. (BIS, 2005; Fanoni, Giudici & Muratori, 2005; Cruz (Ed.), 2004; Alexander (Ed.), 2003).

Scorecard Approach

The scorecard approach models are based on the expert opinions collected using questionnaires (scorecard). By scorecard, the frequency, the severity and the quality of the controls are appraised and so the effectiveness of the system of risk management is already integrated in the model (Alexander (Ed.), 2003).

Inside the questionnaires the frequency and the severity are (generally) classified in five levels (high, high/middle, middle, middle/low and low) and similarly the quality of the controls with (excellent, good, fair, weak and poor). The questionnaire is periodically compiled (every six months, every year, etc…) and, usually, the expert himself is the one who fill it (self-assessment), (Fanoni, Giudici & Muratori 2005; Alexander (Ed.), 2003; Cruz, 2002).

The Basel Committee requires that the estimates are validated on a quantitative base using internal and external historical data related to the OR losses. The difference between this approach and the qualitative method is that the risk profile can change in function of the results that periodically emerge from the compilation of the scorecards. In such way, the method follows the "trend" recognized by the experts (forward-looking characteristic) and so it has a prevision action, (Cruz (Ed.), 2004; Alexander (Ed.), 2003).

The procedure of assessment is obtained after having mapped the activities of the bank and the possible risks according to the standards defined by Basel.

The assessment operation depends on factors as, for instance, (see Alexander (Ed.), 2003):

- The nature of the analyzed activities.
- The geographical location.
- The greatness and complexity of the main activities and of the necessary operations to perform them.

Besides, the choice of the indicators and the appropriate metrics to define the risk profile (as the *Key Risk Driver*) and to monitor and control the harmful events (as the *Key Risk Indicator*) will depend on the previous phase.

The results obtained are visualized on a graph (risk map) where on axles we report for each risk the frequency and the severity. Therefore the graph shows how the various harmful events are distributed and emphasizes possible LFHI or HFLI situations, (Alexander (Ed.), 2003). Afterwards, the graph can be divided in action zones to define the activities to be implemented to face such risks, as, for instance, to accept, to share, to avoid or to transfer them, etc.

Once defined the risk frequencies and severities, some *scenario analyses* are performed to underline the risks, the control systems and the losses associated at the sceneries (see Alexander (Ed.), 2003).

Actuarial and Analytical Models

In these models we use the database of the historical data to estimate either the loss distribution via simulation or the necessary parameters to calculate the VaR, the expected and unexpected losses. The only use of historical data makes these models backward-looking and, therefore, not very flexible for forecasts.

Actuarial Model

The actuarial model calculates the VaR through a percentile of the annual loss distribution (Loss Distribution Approach).

Once we have defined the operational losses database (see Cruz 2002) used for the construction of the actuarial model we have to calculate for every intersection BL/ET the distributions of the frequency (N) and the severity (X). The first distribution is discrete and the second is continuous (see Cruz, 2002; King, 2001).

The loss distribution is evaluated by the combination of the frequency and severity distributions. Then, from the loss distribution we calculate the VaR and, subsequently, the unexpected loss defined as the difference between the VaR and the expected loss.

After we have found the probability density function (p.d.f) for the frequency and the severity, we will combine them by calculating the aggregate loss L:

$$L = \sum_{i=1}^{N} X_i \quad \text{with N~P (n; } \mathbf{a}\text{) and } X_i \sim f(x_i; \mathbf{b});$$

where P(•) and f(•) are respectively the p.d.f of frequency and severity with vector parameters **a** and **b**.

It is very difficult to determine the distribution of L. So under the following hypotheses:

- The severities X_i are identically and independent distributed.
- The distribution of frequency (N) is independent on that of severity (X_i).

Such distributions can be calculated via Monte-Carlo simulation.

By implementing a simulation method we obtain an L realization from which we calculate the VaR, the expected and the unexpected loss (Colombo & Gigli, 2005; Fanoni, Giudici & Muratori, 2005; Cruz (Ed.), 2004; Alexander (Ed.), 2003).

We calculate the realization of L for each BL/ET intersection. It is important to underline that the value of the total VaR, obtained by considering all intersections, can vary according to the dependences among BL/ET, (see Fanoni, Giudici & Muratori, 2005).

The steps for the construction of the actuarial model are summarized in Fanoni, Giudici & Muratori (2005) and in Alexander (Ed.) (2003). For a deepened quantitative approach see Cruz (2002), Cruz (Ed.) (2004) and King (2001).

Analytical Model

The analytical model can be seen as the limit, in a statistical sense, of the actuarial model, when the number of simulations goes to infinity, (Fanoni, Giudici & Muratori, 2005; Alexander (Ed.), 2003). With this model the unexpected loss is calculated (Operational Risk Requirement or ORR), without explicitly determining the loss distribution. A strong limitation, that we have to accept, is the use of Normal distribution for the severity. For the frequencies distribution we can specify every kind of discrete distributions.

The ORR according to the analytical model is:

$$ORR = \phi \times \sigma_L$$

(σ_L = standard deviation of the loss's distribution) and it is calculated with the following formula:

$$ORR = \phi\, \mu_s \sigma_f \sqrt{1 + \left(\frac{\sigma_s}{\mu_s}\right)^2}$$

- σ_f is the frequency standard deviation.
- μ_s and σ_s are respectively the average and standard deviation of the severity.
- ϕ is the skewness coefficient of the loss distribution.

The asymmetry coefficient ϕ is defined as the ratio between the difference of the VaR and the expected loss and the standard deviation of the loss distribution:

$$\phi = \frac{VaR - (\text{expected loss})}{\sigma_L}$$

Due to the difficulty of calculating the shape of the loss's distribution, the coefficient ϕ is difficult to esteem. In every case we can indicate the extreme of the interval for ϕ.

The lower limit depends on the fact that the severity distribution is a Gaussian, so when it is combined with the frequency distribution, the skewness's value will be 3.1 if the VaR is calculated at 99.9 percentile.

Since the loss distribution is less asymmetrical than the distributions of the frequencies and severity, the upper limit of ϕ is that of the frequency distribution because the severity is a Gaussian.

From the skewness's (ϕ) interval width we can have an idea of the parameter's uncertainty. Such length is reduced with the increase of the number of risky events.

What we have mentioned above must be done for every BL\ET, therefore it should be:

$$ORR_{ij} = \phi_{ij}\sigma_{ij}$$

where σ_{ij} is the standard deviation of the annul loss distributions for every intersection BL/ET (i-*th* business line and j-*th* event type).

As for the actuarial approach in function of the level of correlation among the risks for every BL\ET, the ORRs are combined in opportune way (see Alexander (Ed.), 2003). This model cannot be applied in the case of negative correlation, (Fanoni, Giudici & Muratori 2005; Alexander (Ed.), 2003).

Bayesian Approach

The Bayesian approach allows the integration of objective data (generally quantitative type as internal historical loss) with those subjective (generally qualitative type comes from self-assessment or those of external database). In this manner different sources of information can be composed and the model can also undergo to the AMA standards.

The subjective data are used for estimating the prior p.d.f (which can be certain or uncertain) of the parameter of interest, while the objective data are used for estimating the likelihood for the same parameter.

With the Bayesian technique we can esteem the parameters for the distributions of interest (frequency, severity or directly the loss) using the posterior distribution obtained by the combination of the likelihood and prior distributions. In general, the likelihood and the posterior distributions are calculated by using numerical methods as Monte Carlo Markov Chain, (Alexander (Ed.), 2003; Cruz, 2002).

Besides, schemes of inference can be implemented to decide which is the best action to utilize to decrease the loss. In this way we are able to appraise the different strategies of containment, (Cruz, 2002).

In this type of approach there are the Bayesian Networks (BN) which belong to the class of the Probabilistic Expert Systems (see Cowell, Dawid, Lauritzen & Spiegelhalter, 1999). The BN is used to define the loss distribution and to evaluate the strategies of intervention to mitigate the risks; in such case, they take the name of decisional graph, (Alexander (Ed.), 2003; Giudici, 2003; Jensen, 2001). With the BN is performed a multivariate and an integrate analysis of the different kinds of risks or losses. If we use the AMA classification in BL/ET we will have a BN with 56 nodes. In this case every node represents a loss.

A BN is an acyclic graph formed by nodes (or vertexes) and direct arcs. Every vertex represents a casual variable with discrete or continuous states. The relationships among the variables, given by the arcs, are interpreted in terms of conditioned probability according to the Bayes theorem.

The concept of conditioned independence allows the factorization of the joined probability, through Markov property, in a series of local terms which describe the relationships among the variables:

$$f(x_1, x_2, ..., x_n) = \prod_{i=1}^{n} f(x_i \mid pa(x_i))$$

where the pa (x_i) indicates the parents' state of the variables in the nodes X_i.

The problem of a BN is the necessity of a good database to extract the conditioned probabilities and the structure of the net (learning problem) (Jensen, 2001; Heckerman, 1995).

The Bayesian methods allow us to combine the backward-looking with the forward-looking technique; they are useful to consider the dependences among the BL\ET losses and also to appraise the impact of casual factors (or causal depend on the use) on them, (Cornalba & Giudici 2004; Giudici & Bilotta, 2004).

The most evident criticism to these approaches is that the information on the prior distribution can be subjective. However this could be also a good thing if we wanted to make the method sensitive to the personnel of the institution.

FUTURE TRENDS

The Basel Committee foresees the complete realization of the second accord and also of the operational risk standards, for the end of 2007. For the financial institutions that don't consider the operational risks as possible causes of losses yet, it will be opportune to begin to implement an operational loss database where to apply the models.

Besides, the standards for the operational risk will influence not only the banks, but also the access to the credit of the small and medium enterprises which will be subjected to the operational risk evaluation for a better interest rate.

The operational risk is well specified in the banking sector but it is important to consider also other fields, as for example those of telecommunication or manufacture industry and so on. In the future we will see more operational risk analysis on Information Technology (IT) field, especially in the case of IT intensive business (as for online bank). On this road we will observe the birth of IT Operational Risks which will be defined not only for banks but also for different industry fields.

CONCLUSION

Among the different models there is not yet the tendency to use a model more than another, because it strongly depends on the resources of the institution and on its strategy.

The scorecard methods are completely different from those actuarial and analytical because they analyze the institution from different aspects. The scorecards see the vision of the people (therefore they can be used in absence of an appropriated DB), instead the others that of the data. For both, the construction is not simple since the scorecards need good questionnaires and very efficient personnel due to the subjectivity of the procedure, instead the actuarial and analytical models need good database to identify the parameters of the distributions. All methods, however, have problems due to the modeling of the dependences. For the quantitative models the dependences can be considered with the *copula* functions (Dalla Valle, Fantazzini & Giudici 2005; Alexander (Ed.), 2003) and the Bayesian Network, while for the scorecards there are more difficulties but an approach at such problem can be done with BN (Bonafede & Giudici, 2006).

In conclusion, the Bayesian approaches can integrate different sources of information so they are more sensitive to background changes. Moreover, they allow the evaluation of strategies of intervention even if there are more difficulties to be implemented.

REFERENCES

Alexander, C. (Ed.). (2003). *Operational Risk, regulation analysis and management.* London: Prentice Hall.

Bonafede, C. E., & Giudici, P. (2006). *Construction of a bayesian network for a project of enterprise risk management*, (Tech. Rep.). University of Pavia, Italy.

BIS (1988). Basel Committee: International convergence of capital measurement and capital standards. *Basel Committee on Banking Supervision*, January, from http://www.bis.org.

BIS (2001). Basel II: The New Basel Capital Accord - Second Consultative Paper. *Basel Committee on Banking Supervision*, January, from http://www.bis.org.

BIS (2003). Basel II: The New Basel Capital Accord - Third Consultative Paper. *Basel Committee on Banking Supervision*, April, from http://www.bis.org.

BIS (2005). International Convergence of Capital Measurement and Capital Standards. *Basel*

Committee on Banking Supervision, from http://www.bis.org/publ/bcbsca.htm.

Colombo, A., & Gigli, N. L. (2005). Advanced models for the operational risk: loss distribution approach and an application to the analysis of insurance mitigation. *Newsletter AIFIRM, risk management magazine*, 2, pp. 10-24.

Cornalba, C., & Giudici P. (2004). Statistical models for operational risk management. *Physica A*, 338, pp. 166-172.

Cowell, R.G., Dawid, A. P., Lauritzen S.L., & Spiegelhalter D.J. (1999). *Probabilistic Networks and Expert Systems.* New York, USA: Springer.

Cruz, M.G., (2002). *Modelling, measuring and hedging operational risk.* West Sussex, UK: John Wiley and Sons.

Cruz, M.G. (Ed.). (2004). *Operational risk modelling and Analysis.* London, England: Risk Books.

Dalla Valle, L., Fantazzini D., & Giudici P. (2005) (in press). Copulae and Operational Risks. *Journal of Risk Assessment and Management.*

Fanoni, F., Giudici, P., & Muratori, G.M. (2005). *Operational risk: measurement, modelling and mitigation*. Milan, Italy: Il Sole 24 Ore.

Giudici, P. (2003). *Applied Data Mining.* West Sussex, England: John Wiley and Sons.

Giudici, P., & Bilotta A. (2004). Modelling Operational Losses: A Bayesian Approach. *Qual. Reliab. Engng. Int.,* 20, pp. 401-417.

Heckerman, D., (1995). A tutorial on learning with Bayesian networks. *Microsoft Research tech. report MSR-TR-95-06.* Revised November 1996, from http://research.microsoft.com.

Jensen , F.V., (2001). *Bayesian networks and decision graphs.* New York, USA: Springer.

King, J. L., (2001). *Operational risk, Measurement and Modelling.* West Sussex, England: John Wiley and Sons.

KEY TERMS

Backtesting: Statistical hypothesis tests to verify either the VaR or the ORR.

BIA or Basic Indicator Approach: A top-down methodology, for the calculation of the capital requirement (CR_{OR}) to cover OR, based on the average of the gross income (GI) over the previous three years (t):

$$CR_{OR} = \alpha \times \sum_{t=1}^{3} \frac{GI_t}{3} = 15\% \times \sum_{t=1}^{3} \frac{GI_t}{3}$$

where α is a coefficient esteemed by the Basel Committee in a value of 15%.

Copula: An expression for the multivariate distribution in function of the marginal distributions. Given a random vector $\{X_1,\ldots,X_n\}$, its marginal distributions $F_{Xi}(x)$ and the copula function that describes the dependences among the variables, then the $F_{Xi}(x)$ can be combined by copula function to get the joint distribution with the demanded dependence.

Key Risk Driver or KRD: A measure associated to the company risk profile. Meaningful changes of the KRD could implicate important changes on the general level of the quality or to point out a potential increase to the exposure to operational risks or of other nature. In this manner a KRD gives a predictive action.

Key Risk Indicator or KRI: A category of measure used for monitoring the activities and the controls because the KRI has association with the losses; it will be monitored during the year. Such a variable is identified during the phase of process mapping in which the KRDs are identified and then the KRIs. The expert opinions can be weighed by KRI.

Scenario Analysis: A verification of the model sensitivity with the variation of some parameters as frequency, severity and correlation among the losses.

STA or Standard Indicator Approach: Top-down methodology, for the calculation of the capital requirement (CR_{OR}) to cover OR, based on the average of the gross income (GI) of eight business lines (i) over the previous three years (j):

$$CR_{CR} = \sum_{i=1}^{8} \beta_i \times \left(\frac{\sum_{j=1}^{3} \max\{GI_{ij};0\}}{3} \right);$$

where β_i is a coefficient (for every business lines) esteemed by the Basel Committee.

Value at Risk: The possible maximum loss in the 99.99% of the cases in a year, this is the 99.99 percentile of the loss's distribution. Operationally it is used the 99 percentile.

Statistical Web Object Extraction

Jun Zhu
Tsinghua University, China

Zaiqing Nie
Web Search and Mining Group Microsoft Research Asia, China

Bo Zhang
Tsinghua University, China

INTRODUCTION

The World Wide Web is a vast and rapidly growing repository of information. There are various kinds of objects, such as products, people, conferences, and so on, embedded in both statically and dynamically generated Web pages. Extracting the information about real-world objects is a key technique for Web mining systems. For example, the object-level search engines, such as *Libra* (http://libra.msra.cn) and *Rexa* (http://rexa.info), which help researchers find academic information like papers, conferences and researcher's personal information, completely rely on structured Web object information.

However, how to extract the object information from diverse Web pages is a challenging problem. Traditional methods are mainly template-dependent and thus not scalable to the huge number of Web pages. Furthermore, many methods are based on heuristic rules. So they are not robust enough. Recent developments in statistical machine learning make it possible to develop advanced statistical Web object extraction models. One key difference of Web object extraction from traditional information extraction from natural language text documents is that Web pages have plenty of structure information, such as two-dimensional spatial layouts and hierarchical vision tree representation. Statistical Web object extraction models can effectively leverage this information with properly designed statistical models.

Another challenge of Web object extraction is that many text contents on Web pages are not regular natural language sentences. They have some structures but are lack of natural language grammars. Thus, existing natural language processing (NLP) techniques are not directly applicable. Fortunately, statistical Web object extraction models can easily merge with statistical NLP methods which have been the theme in the field of natural language processing during the last decades. Thus, the structure information on Web pages can be leveraged to help process text contents, and traditional NLP methods can be used to extract more features.

Finally, the Web object extraction from diverse and large-scale Web pages provides a valuable and challenging problem for machine learning researchers. To nicely solve the problem, new learning methodology and new models (Zhu et al., 2007b) have to be developed.

BACKGROUND

Web object extraction is a task of identifying interested object information from Web pages. A lot of methods have been proposed in the literature. The wrapper learning approaches like (Muslea et al., 2001; Kushmerick, 2000) take in some manually labeled Web pages and learn some extraction rules (wrappers). Since the learned wrappers can only be used to extract data from similar pages, maintaining the wrappers as Web sites change will require substantial efforts. Furthermore, in wrapper learning a user must provide explicit information about each template. So it will be expensive to train a system that extracts data from many Web sites. The methods (Zhao et al., 2005; Embley et al., 1999; Buttler et al., 2001; Chang et al., 2001; Crescenzi et al., 2001; Arasu, & Garcia-Molina, 2003) do not need labeled training samples and they automatically produce wrappers from a collection of similar Web pages.

Two general extraction methods are proposed in (Zhai & Liu, 2005; Lerman et al., 2004) and they do not explicitly rely on the templates of Web sites. The method in (Lerman et al., 2004) segments data on list

pages using the information contained in their detail pages, and the method in (Zhai & Liu, 2005) mines data records by string matching and also incorporates some visual features to achieve better performance. However, the data extracted by (Zhai & Liu, 2005; Lerman et al., 2004) have no semantic labels.

One method that treats Web data extraction as a classification problem is proposed in (Fin & Kushmerick, 2004). Specifically, Fin & Kushmerick (Fin & Kushmerick 2004) use a support vector machine to identify the start and end tags for a single attribute. For the task of extracting multiple attributes, this method loses the dependencies between different attributes. Instead, the statistical models, which are the theme of this article, can effectively incorporate the statistical dependencies among multiple related attributes, such as a product's name, image, price and description, and achieve globally consistent extraction results.

MAIN FOCUS

Statistical Web object extraction models focus on exploring structure information to help identify interested object information from Web pages. Zhu et al. (Zhu et al, 2005; Zhu et al, 2006; Zhu et al, 2007a; Zhu et al, 2007b) have developed a complete statistical framework for Web object extraction and text content processing on Web pages. The key issues in statistical Web object extraction are selecting appropriate data representation formats, building good graphical models to capture the statistical dependencies, and exploring the structure information of Web pages to process text contents.

Data Representation Formats

Most existing Web mining methods take the HTML source codes or the HTML tag trees as their data representation format. However, due to the low-level representation these methods often suffer from many problems, such as scalability and robustness, when being applied to large-scale diverse Web pages. Instead, the higher level visual information, such as font, position, and size, is more robust and expressive, which will be the features and data representation formats used in statistical models.

Existing statistical models are built based on two different views and data representation formats of Web pages.

1. **2D spatial layout:** When a Web page is displayed to readers, it is actually a two-dimensional image. The HTML elements have their spatial information, such as position (i.e., coordinates in the 2D plane) and size (i.e., height, width, and area). With this spatial information, the elements are well-laid in the 2D plane for easy reading. Thus, the first representation format of Web data is a two-dimensional grid, each of whose nodes represents an HTML element. The edges between the nodes represent the spatial neighborhoods of the HTML elements.
2. **Hierarchical organization:** 2D spatial layout is a flat representation of a Web page. But if we look at the HTML tag trees, which are natural representations of Web pages, we see that the HTML elements are actually pended on a hierarchy. The hierarchical structure in some sense reveals a specific type of organization of the elements. This hierarchical organization information can be helpful in identifying the boundaries of data records or even in identifying the target attributes.

Statistical Web Object Extraction Models

According to the two different data representation formats—2D spatial layout and hierarchical organization, two types of statistical models have been studied.

1. **2D local model:** The two-dimensional Conditional Random Fields (Zhu et al, 2005) are introduced to model a data record, which consists of a set of HTML elements. The 2D model put the elements on a grid according to their spatial information and neighborhood relationships. Each node on the grid is associated with a random variable, which takes values from a set of class labels, such as name, image, price, and description in product information extraction. The model defines a joint distribution of all the variables and we can do some statistical inference to get the most probably labeling results of all the random variables. From the labeling results, we know which element is

the product's name, which is the image, which is the price, and etc. When building the model, we have to provide a set of labeled training data to learn its parameters.

2. **Hierarchical integrated model:** The previous 2D local model is a de-coupled method, that is, the detection of data records and the labeling of HTML elements are performed separately. This will cause several disadvantages, such as error propagation, lack of semantics in data record detection, lack of mutual interactions in attribute labeling, and failure in incorporating long-distance dependencies. To address these problems, hierarchical integrated models (Zhu et al, 2006) are introduced. These models are based on the hierarchical representation of Web pages. However, in practice, instead of the HTML tag trees, vision trees (Cai et al, 2004) are used as the data representation format. Vision trees have their advantages in representing content structure, while HTML tag trees tend to represent presentation structure. Also, tags trees tend to be diverse in the Web environment.

Statistical Models for Text Content Processing

In both the 2D local model and the hierarchical integrated model, HTML elements are taken as atomic extraction units. However, in many cases an HTML element can contain multiple interested attributes. For example, an element in a paper record can contain the names of several coauthors (Zhu et al, 2007a). If we want to identify each author's name, text content processing like segmentation must be performed.

Traditional NLP methods can address this problem in natural language sentences. However, they are not directly applicable to Web data extraction because on Web pages many text contents are text fragments and they are typically lack of the grammars as required in NLP systems. Fortunately, all the proposed statistical models can be easily merged with statistical NLP methods. One preliminary work is (Zhu et al, 2007a), where the structure information of Web pages is integrated into text content processing. Promising results are achieved in extracting researchers' information, such as name, title, affiliation, contact information, and publication information.

FUTURE TRENDS

Statistical Web object extraction techniques will become key components of any Web data mining systems. It is of great interest for machine learning researchers to develop more powerful and more automatic learning systems to extract the huge information from the Web. In the near future, more theories and machine learning techniques of statistical Web object extraction will be developed. Meanwhile, comprehensive comparisons of these theories and techniques will be carried out. More and more object-level vertical search engines will be available for users.

CONCLUSION

Extracting Web object information from diverse Web pages is important and challenging. Traditional approaches like wrapper learning and single attribute extractors cannot satisfy the increasing demands, especially in developing object-level vertical search engines. Statistical machine learning methods are more robust and scalable. To build a good statistical extraction model, selecting an appropriate data representation and building a good graphical model are key steps. On Web pages plenty of structure information, such as two-dimensional spatial layouts and hierarchical vision-tree representation, can be explored in statistical models to achieve promising results.

REFERENCES

Arasu, A., & Garcia-Molina, H. (2003). Extracting structured data from Web pages. In *Proceeding of the International Conference on Management of Data (SIGMOD03)*, 337- 348.

Bunescu, R. C., & Mooney, R. J. (2004). Collective information extraction with relational Markov networks. In *Proceeding of 42nd Annual Meeting of the Association for Computational Linguistics (ACL04)*, 438--445.

Buttler, D., Liu, L., & Pu, C. (2001). A fully automated object extraction system for the world wide Web. In *Proceeding of IEEE International Conference on Distributed Computing Systems*, 361-370.

Cai, D., Yu, S., Wen, J.-R. & Ma, W.-Y. (2004). Block-based Web search. In *Proceeding of International Conference on Information Retrieval (SIGIR04)*, 456-463.

Chang, C.-H., & Liu, S.-L. (2001). IEPAD: information extraction based on pattern discovery. In *Proceeding of the International World Wide Web Conference (WWW01)*, 681-688.

Cohen, W. W., & Sarawagi, S. (2004). Exploiting dictionaries in named entity extraction: combining semi-Markov extraction processes and data integration methods. In *Proceeding of the International Conference on Knowledge Discovery and Data Mining (SIGKDD04)*, 89-98.

Cowell, R. G., Dawid, A. P., Lauritzen, S. L., & Spiegelhalter, D. J. (1999). *Probabilistic networks and expert systems*. Springer.

Crescenzi, V., Mecca, G., & Merialdo, P. (2001). ROADRUNNER: towards automatic data extraction from large Web sites. In *Proceeding of the Conference on Very Large Data Base (VLDB01)*, 109-118.

Embley, D. W., Jiang, Y., & Ng, Y.-K. (1999). Record-boundary discovery in Web documents. In *Proceeding of the International Conference on Management of Data (SIGMOD*99), 467-478.

Finn, A., & Kushmerick, N. (2004). Multi-level boundary classification for information extraction. In *Proceeding of European Conference on Machine Learning*, 111-122.

Kushmerick, N. (2000). Wrapper induction: efficiency and expressiveness. *Artificial Intelligence*, 118:15-68.

Lerman, K., Getoor, L., Minton, S., & Knoblock, C. (2004). Using the structure of Web sites for automatic segmentation of tables. In *Proceeding of the International Conference on Management of Data (SIGMOD04)*, 119-130.

Lerman, K., Minton, S., & Knoblock, C. (2003). Wrapper maintenance: a machine learning approach. *Journal of Artificial Intelligence Research*, 18:149-181.

Muslea, I., Minton, S., & Knoblock C. A. (2001). Hierarchical wrapper induction for semi-structured information sources. *Autonomous Agents and Multi-Agent* 4, 1/2, 93-114.

Nahm, U. Y., & Mooney, R. J. (2001). A mutually beneficial integration of data mining and information extraction. In *Proceeding of the Conference on Artificial Intelligence (AAAI01)*, 627-632.

Sarawagi, S., & Cohen, W. W. (2004). Semi-Markov conditional random fields for information extraction. *Advances in Neural Information Processing Systems (NIPS04)*.

Skounakis, M., Craven, M., & Ray S. (2003). Hierarchical hidden Markov models for information extraction. In *Proceeding of the International Joint Conference on Artificial Intelligence (IJCAI03)*, 427-433.

Wellner, B., McCallum, A., Peng, F., & Hay, M. (2004). An integrated conditional model of information extraction and coreference with application to citation matching. In *Proceeding of the Conference on Uncertainty in Artificial Intelligence (UAI*04), 593-601.

Zhai, Y., & Liu, B. (2005). Web data extraction based on Partial tree alignment. In *Proceeding of the Conference on World Wide Web (WWW05)*, 76-85.

Zhao, H., Meng, W., Wu, Z., Raghavan, V., & Yu, C. (2005). Fully automatic wrapper generation for search engines. In *Proceeding of the Conference on World Wide Web (WWW05)*, 66-75.

Zhu, J., Nie, Z., Wen, J.-R., Zhang, B., & Ma, W-Y. (2005). 2D conditional random fields for web information extraction, In *Proceeding of the International Conference on Machine Learning (ICML05)*, 1044-1051.

Zhu, J., Nie, Z., Wen, J.-R., Zhang, B., & Ma, W-Y. (2006). Simultaneous record detection and attribute labeling in web data extraction. In *Proceeding of the International Conference on Knowledge Discovery and Data Mining (SIGKDD06)*, 494-503.

Zhu, J., Nie, Z., Wen, J.-R., Zhang, B., & Hon, H-W. (2007a). Webpage understanding: an integrated approach. In *Proceeding of the International Conference on Knowledge Discovery and Data Mining (SIGKDD07)*, 903-912.

Zhu, J., Nie, Z., Zhang, B., & Wen, J.-R. (2007b). Dynamic hierarchical Markov random fields and their application to web data extraction. In *Proceeding of the International Conference on Machine Learning (ICML07)*, 1175-1182.

KEY TERMS

Conditional Random Fields: One type of graphical models that define a conditional probability of labels given the input features. The advantage of this model is that it does not pay efforts to model the input features and thus no independence assumption is made. In contrast, the generative models like HMMs directly model the input features and define a joint distribution of both labels and inputs. To achieve computational tractability, strong independence assumption must be made in generative models.

Data Records: Another data unit of Web pages. A data record is a block in a Web page and contains the information of a single object. For example, a product record usually contains the product's name, image, price, and some descriptions.

Graphical Models: A modeling language that combines graph theory and probability theory. A graphical model consists of two parts—a graph and a probability distribution of the random variables associated with the graph's vertices. Given a graphical model, various queries can be asked, such as the marginal probability, the posterior probability, and the most likely assignment of some random variables. The inference can be efficiently carried out by exploring the sparseness of the graph structure.

HTML Elements: The atomic data units of Web pages. An HTML element can be a text fragment, an image, a button, or just some decoration items.

Statistical Web Object Extraction: Web object extraction models that learn statistical information from provided data sets. It can effectively incorporate heterogeneous features of Web data.

Structured Extraction Model: One type of extraction models in which multiple attributes are extracted at the same time. For example, for product information extraction all the product attributes (e.g. name, image, price, and description) are identified simultaneously. The advantage of structured extraction models is that they can incorporate the statistical dependencies between multiple attributes, which will be lost in a model that identifies each attribute separately.

Vision Tree: A data representation format of Web pages. Like HTML tag trees, a vision tree is itself a tree. The difference from HTML tag trees is that tag trees tend to reveal presentation structure rather than content structure and are often not accurate enough to discriminate different semantic parts in a Web page. A vision tree is built with a vision-based page segmentation (VIPS) approach (Cai et al., 2004), which makes use of page layout features such as font, color, and size to construct the tree. Each node on a vision tree represents a data block on a Web page. The root block represents the whole page and each inner block is the aggregation of all its child blocks. The leaf blocks are the atomic HTML elements, which form a flat segmentation of the Web page.

Web Object Extraction: An information extraction task that identifies interested object information from Web pages. The difference from traditional information extraction is that traditionally information extraction is typically conducted on natural language documents which are quite different from Web pages. Natural language sentences are composed according to a particular type of grammars, while the HTML elements on Web pages have more structures but are typically lack of grammars.

Storage Systems for Data Warehousing

Alexander Thomasian
New Jersey Institute of Technology - NJIT, USA

José F. Pagán
New Jersey Institute of Technology - NJIT, USA

INTRODUCTION

Data storage requirements have consistently increased over time. According to the latest WinterCorp survey (http://www/WinterCorp.com), "The size of the world's largest databases has tripled every two years since 2001." With database size in excess of 1 terabyte, there is a clear need for storage systems that are both cost effective and highly reliable.

Historically, large databases are implemented on mainframe systems. These systems are large and expensive to purchase and maintain. In recent years, large data warehouse applications are being deployed on Linux and Windows hosts, as replacements for the existing mainframe systems. These systems are significantly less expensive to purchase while requiring less resources to run and maintain.

With large databases it is less feasible, and less cost effective, to use tapes for backup and restore. The time required to copy terabytes of data from a database to a serial medium (streaming tape) is measured in hours, which would significantly degrade performance and decreases availability. Alternatives to serial backup include local replication, mirroring, or geoplexing of data.

The increasing demands of larger databases must be met by less expensive disk storage systems, which are yet highly reliable and less susceptible to data loss.

This article is organized into five sections. The first section provides background information that serves to introduce the concepts of disk arrays. The following three sections detail the concepts used to build complex storage systems. The focus of these sections is to detail: (i) *Redundant Arrays of Independent Disks* (RAID) arrays; (ii) multilevel RAID (MRAID); (iii) concurrency control and storage transactions. The conclusion contains a brief survey of modular storage prototypes.

BACKGROUND

The fifty year old magnetic disk drive [technology] remains a viable storage medium because they can accommodate an excess of 500 Gigabytes (GB), are nonvolatile, inexpensive, have an acceptable random access time (10 milliseconds), and exhibit a *Mean Time to Failure (MTTF)* exceeding 10^6 hours. Concurrent with their benefits, disk failures occur frequently in large data centers.

RAID serves to mitigate disk failures in large installations (Chen et al. 1994). RAID level 5 (RAID5) masks the failure of a single disk by reconstructing requested blocks of a failed disk, on demand. Additionally, it automatically reconstructs the contents of the failed disk on a spare disk.

The RAID paradigm is inadequate for *Very Large Disk Arrays (VLDA's)* used in data warehousing applications, because the non-disk components may be less reliable than the physical disks. *Hierarchical RAID (HRAID)* achieves a high reliability by using multiple levels of RAID controllers (Baek et al. 2001). The higher and lower RAID levels of HRAID are specified as RAIDX(M)/Y(N), where X and Y are the RAID level and *M* and *N* denote the number of virtual disks or *storage nodes (SN's)* at the higher level and physical disks at the lower level. *Multilevel RAID (MRAID)* was proposed as an alternative to HRAID, because of its two key differences: (i) disks are organized into *SN's* (or bricks) at the lower level, this constitutes the *smallest replaceable unit (SRU)*, (ii) the association among SN's is logical and dynamic rather than hardwired (Thomasian 2006).

Each SN consists of an array of disks, an array controller, a partially nonvolatile cache, and the capability to interconnect. SN costs are kept low, in some designs, by using mirroring to protect data on each SN.

The internal structure of a brick in IBM's Intelligent Brick prototype is illustrated in (Wilcke et al. 2006)

Figure 2. Bricks are cube shaped and communicate via capacitive coupling between insulated flat metal plates at each of its six surfaces. Higher capacities are attained by stacking bricks on top of each other. Gigabit Ethernet is used to provide connectivity to external cubes. A fail-in-place or deferred maintenance system is also postulated.

MAIN THRUST

We first review RAID systems, before discussing multilevel RAID. We next describe storage transactions, which are required for the correct operation of the system.

RAID LEVELS

There are seven RAID levels (Chen et al. 1994) which are classified as *k disk failure tolerant (kDFT)* arrays (Thomasian et al. 2007). RAID0 is 0DFT because it has no redundancy. Data is divided into *striping units (SUs)*, which are written to the disks in a round-robin manner. This technique of writing data, in small units, across multiple disks is termed striping. Striping also serves to balance the load across an array of *N* disks.

RAID1 is also referred to as *Basic mirroring (BM)*. The simplest RAID1 configuration requires two disks, where data is written such that both disks are exact copies of each other. RAID1 has *M = N/2* disk pairs, where *N* is the number of physical disks. RAID1 can tolerate up to M disk failures (one disk of a pair), however failure of both disks (in a pair) will lead to data loss. Hence, RAID1 is 1DFT. BM has twice the read transaction rate of a single disk because reads are duplexed between the disk pairs. Similarly, a single disk failure will shift the load to the working disk and essentially double the load on the surviving disk.

In *Interleaved Declustering (ID)* the total number of disks (*N)* are divided into *c* groups, so that there are *M = N / c* disks per group. In this scheme, disks are partitioned into primary and secondary areas. Data is written completely to a primary area of a single disk. Additionally, this data is divided into blocks and written to the secondary areas of the remaining *M – 1* disks in the group. Each group is a 1DFT array and a single disk failure will increase the load on the surviving disks by *M / (M – 1)*.

In *Chained Declustering (CD)* the primary data on each disk is replicated to the secondary area of the next disk modulo *N*, e.g., for $i < N$ $(N+i) \bmod N = i$.

In *Group Rotate Declustering (GRD)*, data is striped across a group of disks (*M = N / 2)* and replicated (in a rotated manner) across the remaining *M* disks. The load on surviving disks in CD and GRD may be balanced with appropriate routing of read requests. The tradeoff between load balancing and reliability in these RAID1 configurations is discussed in (Thomasian and Blaum 2006). The aforementioned paradigms can be applied to bricks and to arrays of multiple bricks, thus facilitating a balanced load on failures.

RAID2 uses Hamming codes which are inefficient and increases the number of check disks in proportion to the logarithm of existing data disks.

RAID3, RAID4, and RAID5 are 1DFTs because they require one disk for parity. RAID3 uses small SUs which are intended for parallel data transfers, thus increasing the transfer rate. RAID3 is most efficient with synchronized disks. RAID4 allocates the parity SU on the Nth disk which is computed from the SUs on the first *N-1* disks. On disk *N*, let D_i, $1 \le i \le N-1$ denote the first *N-1* SUs. The parity of the SUs is: $P_{1:4} = D_1 \oplus D_2 \oplus D_3 \oplus D_4$ where $\oplus$ is the XOR operator, such that $1 \oplus 1 = 0$. If disk 1 fails then block d_1 in SU D_1 can be reconstructed as $d_1 = d_2 \oplus d_3 \oplus d_4 \oplus p_{1:4}$.

When data block d_1 is updated, instead of reading all corresponding blocks, we can update the parity block by computing $d_1^{diff} = d_1^{old} \oplus d_1^{new}$ and using d_1^{diff} to compute $p_{1:4}^{new} = p_{1:4}^{old} \oplus d_1^{diff}$. RAID4 has two disadvantages: (i) the parity disk is not accessed by read requests, (ii) it becomes a hot-spot when write requests dominate. RAID5 uses the left-symmetric distributed parity layout. Here, blocks are arranged in left to right diagonals. Reading a block on a failed disk, in RAID5, will generate reads to corresponding blocks on the surviving *N - 1* disks, thus doubling the load on these disks. A solution to this problem is *clustered RAID – CRAID*, which uses a parity group of size *G < N*, so that load increase to the other disks is $\alpha = (G - 1)/(N - 1)$.

RAID6 is a 2DFT with two parity disks, P and Q parities. Reed-Solomon codes and specialized parity codes: EVENODD, *Row Diagonal Parity - RDP*, X-codes, and RM2 (Thomasian and Blaum 2007) are typically used. X-codes require *N* to be prime, while

RM2 uses more than the minimal capacity of two disks for parities. In a 2DFT array (such as RAID6) incurs a higher processing overhead than a 1DFT (such as RAID5) because writes must update two parities instead of one. The relative performance of RAID6 and RM2 versus RAID5 and RAID0 is investigated in (Thomasian et al. 2007).

Rebuild is a systematic reconstruction of a failed disk on a spare. Successive tracks of the surviving disks are read and XOR-ed to compute missing tracks. Distributed sparing is a better alternative because it efficiently uses the bandwidth of all disks. If the parity SUs in RAID5 are used as spare SUs, then RAID5 reverts to a RAID0. However, If the Q parity of RAID6 is used as a spare SU, then RAID6 reverts to a RAID5.

Internode Replication and Erasure Coding

Replication is preferable to erasure coding because it attains a higher access rate. Additional improvement can be realized by judicious routing and scheduling of requests, e.g., the *distributed shortest processing time first – DSPTF* method (Lumb, Golding, and Ganger 2004).

Read requests can be processed by any SN holding the requested data, however, updates should be applied to all SNs, i.e., read-one, write-all paradigm. The linearizability correctness paradigm requires reads to access the latest version of any data. More generally, appropriate *concurrency control (CC)* algorithms are required for replicated data (Thomasian 1996).

Erasure coding across SNs is discussed in (Thomasian 2006). If the number of SNs in a VLDA is very large, erasure coding may be applied to SN clusters. Consider *M* SNs in a cluster, which is organized as a Multilevel RAID array, that is *l node failure tolerant* (*l NFT)*. Each SN, then, will contain *N* disks and is a *k*DFT array. For simplicity, we will use $k = l = 1$, i.e., nested RAID arrays written as RAID5(M)/5(M).

It would be inefficient to allocate one SN for all of the Q parities. Instead, distributing the Q parities among all SNs would better balance the read load. Similarly, each SN can allocate *1/N* of its capacity to P parities and another *1/N* to Q parities, and the two parities can be placed in parallel diagonals to balance the load. P parities will protect data blocks (and Q parities) in the same stripe, while Q parities serve to protect data blocks across SNs.

Example 1

A disk array composed of four SNs containing four disks each, where $M = N = 4$, is shown in Table 1. Each SU is designated as $x^{m}_{i,j}$, where $1 \le i \le N$ is the stripe or row number, $1 \le j \le N$ is the disk number, and $1 \le m \le M$ is the SN number (*x* may be either *d*, *p*, or *q*). The Q parities are rotated across SNs and disks, to balance the load.

To update $d^{2}_{4,1}$ we:

1. Compute $d^{2diff}_{4,1} = d^{2new}_{4,1} \oplus d^{2old}_{4,1}$ and update its parities in parallel:
$$p^{2new}_{4,3} = p^{2old}_{4,3} \oplus d^{2diff}_{4,1}, q^{1new}_{4,1} = q^{1old}_{4,1} \oplus d^{2diff}_{4,1}.$$
2. Compute $q^{1diff}_{4,1} = q^{1new}_{4,1} \oplus q^{1old}_{4,1}$ to update its *p* parity: $p^{1new}_{4,4} = p^{1old}_{4,4} \oplus q^{1diff}_{4,1}$.

Disk failures can be dealt with either within an SN, or with an erasure coding scheme across SNs. For example, consider a failure of SN_1 where we need to reconstruct its first stripe:

$$d^{1}_{1,1} = d^{2}_{1,1} \oplus q^{4}_{1,1}$$

$$d^{1}_{1,2} = q^{3}_{1,2} \oplus d^{4}_{1,2}$$

$$q^{1}_{1,4} = d^{2}_{1,4} \oplus d^{3}_{1,4}$$

$$p^{1}_{1,3} = d^{1}_{1,1} \oplus d^{1}_{1,2} \oplus q^{1}_{1,4}$$

If disk 1 at SN_2 had failed, it should be reconstructed before SN_1 can be reconstructed:

$$d^{2}_{1,1} = p^{2}_{1,2} \oplus q^{2}_{1,3} \oplus d^{2}_{1,4}$$

For higher efficiency, it is better to overlap the rebuilding of disk 1 at SN_2 with the rebuilding SN_1.

Storage Transactions

A race condition arises when updating $p^{1}_{1,3}$ to reflect changes to both $d^{1}_{1,1}$ and $d^{1}_{1,2}$, at the same time. The

Table 1. Data layout in MRAID5(4)/5(4) - nested RAID5 arrays with M = 4 SNs and N = 4 disks per SN.

Node 1				Node 2				Node 3				Node 4			
$d^1_{1,1}$	$d^1_{1,2}$	$p^1_{1,3}$	$q^1_{1,4}$	$d^2_{1,1}$	$p^2_{1,2}$	$q^2_{1,3}$	$d^2_{1,4}$	$p^3_{1,1}$	$q^3_{1,2}$	$d^3_{1,3}$	$d^3_{1,4}$	$q^4_{1,1}$	$d^4_{1,2}$	$d^4_{1,3}$	$p^4_{1,4}$
$d^1_{2,1}$	$p^1_{2,2}$	$q^1_{2,3}$	$d^1_{2,4}$	$p^2_{2,1}$	$q^2_{2,2}$	$d^2_{2,3}$	$d^2_{2,4}$	$q^3_{2,1}$	$d^3_{2,2}$	$d^3_{2,3}$	$p^3_{2,4}$	$d^4_{2,1}$	$d^4_{2,2}$	$p^4_{2,3}$	$q^4_{2,4}$
$p^1_{3,1}$	$q^1_{3,2}$	$d^1_{3,3}$	$d^1_{3,4}$	$q^2_{3,1}$	$d^2_{3,2}$	$d^2_{3,3}$	$p^2_{3,4}$	$d^3_{3,1}$	$d^3_{3,2}$	$p^3_{3,3}$	$q^3_{3,4}$	$d^4_{3,1}$	$p^4_{3,2}$	$q^4_{3,3}$	$d^4_{3,4}$
$q^1_{4,1}$	$d^1_{4,2}$	$d^1_{4,3}$	$p^1_{4,4}$	$d^2_{4,1}$	$d^2_{4,2}$	$p^2_{4,3}$	$q^2_{4,4}$	$d^3_{4,1}$	$p^3_{4,2}$	$q^3_{4,3}$	$d^3_{4,4}$	$p^4_{4,1}$	$q^4_{4,2}$	$d^4_{4,3}$	$d^4_{4,4}$

value of $p^1_{1,3}$ might be set to either $p^{1new}_{1,3} = d^{1old}_{1,1} \oplus d^{1new}_{1,1} \oplus p^{1old}_{1,3}$ or $p^{1new}_{1,3} = d^{1old}_{1,2} \oplus d^{1new}_{1,2} \oplus p^{1old}_{1,3}$, however, $p^1_{1,3}$ should reflect the outcome of both changes; $p^{1new}_{1,3} = p^{1old}_{1,3} \oplus d^{1diff}_{1,1} \oplus d^{1diff}_{1,2}$. A similar situation arises when updating $q^4_{1,4}$ to reflect simultaneous changes to $d^1_{1,1}$ and $d^1_{1,2}$. This is termed the lost update problem.

The lost update problem can be prevented by encapsulating each update inside a transaction. Let R and W represent reading and writing of the data block in parentheses. Read requests obtain a shared lock on the object being read. The lock is promoted to an exclusive lock when the object is updated. According to strict *two-phase locking (2PL)*, locks are held until the transaction commits using a *two-phase commit (2PC)* protocol, however, a less sophisticated protocol may be appropriate for this case, (example in Thomasian 1996) and (Ramakrishnan and Gehrke (2003)).

Transaction (update $d^{1new}_{1,2}$) = {

1. $R(d^{1old}_{1,1})$
2. $W(d^{1new}_{1,1})$
3. $d^{1diff}_{1,1} = d^{1new}_{1,1} \oplus d^{1old}_{1,1}$
4. $R(p^{1old}_{1,3}), R(q^{4old}_{1,1})$
5. $p^{1new}_{1,3} = d^{1diff}_{1,1} \oplus p^{1old}_{1,3}, q^{4new}_{1,1} = d^{1diff}_{1,1} \oplus q^{4old}_{1,1}, W(p^{1new}_{1,3})$
6. $q^{4diff}_{1,1} = q^{4new}_{1,1} \oplus q^{4old}_{1,1}, R(p^{4old}_{1,4}),$
7. $p^{4new}_{1,4} = p^{4old}_{1,4} \oplus q^{4diff}_{1,1}$
8. $W(p^{4new}_{1,4})$

}.

If disk 1 at SN_1 fails, rather than aborting the transaction, we reconstruct $d^{1old}_{1,1} = d^1_{1,2} \oplus p^1_{1,3} \oplus q^1_{1,4}$ via a fork-join request. This requires the locking of accessed objects to reconstruct $d^1_{1,1}$, that is, $d^1_{1,2}$ and $p^1_{1,3}$ (note that $d^1_{1,2}$ is also being updated). Similarly, there is no need to abort the transaction if disk 3 at SN_1 has failed (making $p^1_{1,3}$ unavailable) because the parity will be reconstructed via the rebuild process. In effect we postulate nested transactions such that an SN with failed disks can issue an appropriate sub-transaction to reconstruct the data. *Base Storage Transactions (BSTs)* proposed in (Amiri et al. 2000) uses a different semantic, where, reading from a failed disk results in an abort and "the parent task" will issue an appropriate BST for access in degraded mode.

Distributed dynamic locking with blocking may lead to distributed deadlocks, whose detection and resolution is costly. Lock conflicts can be resolved at the local level with wound-wait and wait-die locking methods or the time-stamp ordering method (see Thomasian 1996).

Storage Brick Projects

In IBM's Collective Intelligent Brick (CIB) project, the connectivity of a 3-D mesh-based data cube is treated

as a percolation problem (Fleiner et al. 2006). Simulation is used to determine the fraction of failures that renders a system unusable. A 3D (dimensional) mesh is operational as long as 70%of the bricks are working, while a 2D mesh requires 85% of bricks because fewer alternate paths are available (Wilcke et al. 2006).

Rao et al. (2005) describe an array of bricks, where each brick is one of RAID0, RAID5, or RAID6, and erasure coding is implemented across bricks. The performance metric of interest is "data loss events per petabyte-year'' (petabyte 10^{12}). The sensitivity of *Mean Time to Data Loss (MTTDL)* to MTTFs, rebuild block size, and link speed is investigated.

Federated Array of Bricks (FAB) is a distributed disk array, implemented at HP Labs, that attains the same reliability as enterprise disk arrays, at a lower cost. As described in Saito et al. (2004) FAB consists of a collection of bricks with a reconfiguration algorithm to move data when bricks are added or decommissioned. A novel feature of FAB is its majority voting based algorithm to replicate or erasure-code logical blocks across disks. This protocol allows linearizable accesses to replicated data, transparently bypassing failed bricks. RepStore, described by Zhang et al. (2004), is a prototype at Microsoft Research in China, which offers both replication and erasure coding. The choice is made at volume creation time in FAB, while RepStore makes the choice adaptively.

A comprehensive review of storage systems is given by Kenchmana-Hosekote (2004). Table 2 provides a classification of data paths, serialization, and atomicity. Serialization is attained by locking.

Xin (2005) studied the reliability mechanisms in very large storage systems with repair. Markov chain models are used to study the *Mean Time to Data Loss (MTTDL)* in 2- and 3-way mirroring and mirrored RAID5 arrays.

FUTURE TRENDS

While there are experimental brick systems: (i) *CIB* at IBM; (ii) *FAB* at HP; and (iii) RepStore at Microsoft Research in China, they are all still in the prototyping stage. It is expected that some of this projects will turn into products, but new software will be needed to make bricks in data warehousing a reality.

CONCLUSION

We have introduced design issues related to arrays of bricks. Multilevel RAID allows each SN to have an internal reliability mechanism, but the RAID paradigm is implemented across SNs to deal with failures.

REFERENCES

Amiri, K., Gibson, G. A., & Golding, R. (2000). Highly concurrent shared storage. *Proceedings 20th International Conference Distributed Computing Systems* (pp. 298-307). Taiwan.

Baek, S. H., Kim B. W., Jeung, E., & Park, C. W. (2001). Reliability and performance of hierarchical RAID with multiple controllers. *Proceedings 20th Annual ACM Symposium on Principles of Distributed Computing* (pp. 246-254). USA.

Chen, P. M., Lee, E. K., Gibson, G. A., Katz, R. H., & Patterson, D. A. (1994). RAID: High-performance, reliable secondary storage. *ACM Computing Surveys, 26*(2), 145-185.

Fleiner, C. et al. (2006). Reliability of modular mesh-connected intelligent storage brick systems. *IBM J. R&D* 50(203), 199-208.

Kenchmana-Hosekote, D. R., Golding, R. A., Fleiner, C., & Zaki, O. A. (2004). The design and evaluation of network RAID protocols. *IBM Research Report RJ 10316*, Almaden, CA.

Lumb, C. R., Golding, R., & Ganger, G. R. (2004). D-SPTF: Decentralized request distribution in brick-based storage systems. *Proceedings 11th International Conference Architectural Support for Programming Languages and Operating Systems - ASPLOS* (pp. 37-47).

Rao, K. K., Hafner, J. L., & Golding, R. A. (2005). Reliability for networked storage nodes. *IBM Research Report RJ 10358*, Almaden, CA.

Thomasian, A. (1996). *Database Concurrency Control: Methods, Performance, and Analysis*. Kluwer Academic Publishers.

Thomasian, A. (2006). Multi-level RAID for Very Large Disk Arrays – VLDAs. *ACM Performance Evaluation Review*, 33(4), 17-22.

Thomasian, A. & Blaum, M. (2006). Mirrored disk reliability and performance. *IEEE Transactions Computers*, 55(12), 1640 -1644.

Thomasian, A. & Blaum. M. (2007). Two disk failure tolerant disk arrays: Organization, operation, coding, & performance analysis. *ACM Computing Surveys*, revised and resubmitted.

Thomasian, A., Han, C., & Fu, G. (2007). Performance of Two-Disk Failure-Tolerant Disk Arrays. *IEEE Transactions Computers*, 56(6), 1-16.

Wilcke, W. W. et al. (n.d.). IBM intelligent bricks project -- Petabytes and beyond. *IBM Journal Research and Development*, 50(2/3), 181-197.

Zhang, Z., Lin, S., Lian, Q., & Jin, C. (2004). Repstore: A self-managing and self-tuning storage backend with smart bricks. *Proceedings First International Conference on Autonomic Computing - ICAC'04* (pp.122-129).

KEY TERMS

Erasure Coding (EC): Erasure coding allows the reconstruction of the contents of a failed disk, whose position is known. Parity and Reed-Solomon codes are forms of EC.

Hierarchical RAID (HRAID): Higher level RAID controllers control lower level RAID controllers, which control disks. This scheme generalizes to more than two levels.

kDFT: *k* disk failure tolerant array.

Multilevel RAID (MRAID): It is similar to HRAID, except multiple RAID arrays are associated to each other logically, rather than physically. An examples is RAIDX(M)/RAIDY(N), where X/Y denote the RAID levels, and M/N denote the number of logical/physical disks at the higher/lower levels, e.g., RAID1(2)/RAID5(7) is two mirrored RAID5s with 7 disks each.

Parity Block: The bits of a parity block are computed by performing an exclusive OR between the corresponding bits of data blocks at the other disks. EVENODD and RDP are parity based schemes, which compute two check blocks using symbols of appropriate size. They are more efficient than Reed-Solomon codes from the viewpoint of exclusive-OR operations.

RAID: Redundant array of independent disks, which is achieved using replication or erasure coding.

Rebuild: Reconstructing the contents of a failed disk on a spare disk, while surviving disks process external requests.

Storage Node (SN or Brick): A self-contained collection of disks, a controller, a DRAM cache, which constitute the smallest replaceable unit. SNs are connected to each other, a power supply, and cooling system.

Storage Transactions: Ensure the correct updating of parity blocks as data blocks are being updated.

Striping: Partition a dataset into equal sized stripe units allocated in round-robin manner across the disks.

Subgraph Mining

Ingrid Fischer
University of Konstanz, Germany

Thorsten Meinl
University of Konstanz, Germany

INTRODUCTION

The amount of available data is increasing very fast. With this data, the desire for data mining is also growing. More and larger databases have to be searched to find interesting (and frequent) elements and connections between them. Most often the data of interest is very complex. It is common to model complex data with the help of graphs consisting of nodes and edges that are often labeled to store additional information. Having a graph database, the main goal is to find connections and similarities between its graphs. Based on these connections and similarities, the graphs can be categorized, clustered or changed according to the application area. Regularly occurring patterns in the form of subgraphs —called *fragments* in this context—that appear at least in a certain percentage of graphs, are a common method to analyze graph databases. The actual occurrence of a fragment in a database graph is called *embedding*. Finding the fragments and their embeddings is the goal of subgraph mining described in detail in this chapter.

The first published graph mining algorithm, called Subdue, appeared in the mid-1990s and is still used in different application areas and was extended in several ways. (Cook & Holder, 2000). Subdue is based on a heuristic search and does not find all possible fragments and embeddings. It took a few more years before more and faster approaches appeared. In (Helma, Kramer, & de Raedt, 2002) graph databases are mined for simple paths, for a lot of other applications only trees are of interest (Rückert & Kramer, 2004). Also Inductive Logic Programming (Finn et al., 1998) was applied in this area. At the beginning of the new millennium finally more and more and every time faster approaches for general mining of graph databases were developed that were able to find all possible fragments. (Borgelt & Berthold, 2002; Yan & Han, 2002; Kuramochi & Karypis, 2001; Nijssen & Kok, 2004).

Several different application areas for graph mining are researched. The most common area is mining molecular databases where the molecules are displayed by their two-dimensional structure. When analyzing molecules it is interesting to find patterns that might explain why a certain set of molecules is useful as a drug against certain diseases (Borgelt & Berthold, 2002). Similar problems occur for protein databases. Here graph data mining can be used to find structural patterns in the primary, secondary and tertiary structure of protein categories (Cook & Holder, 2000).

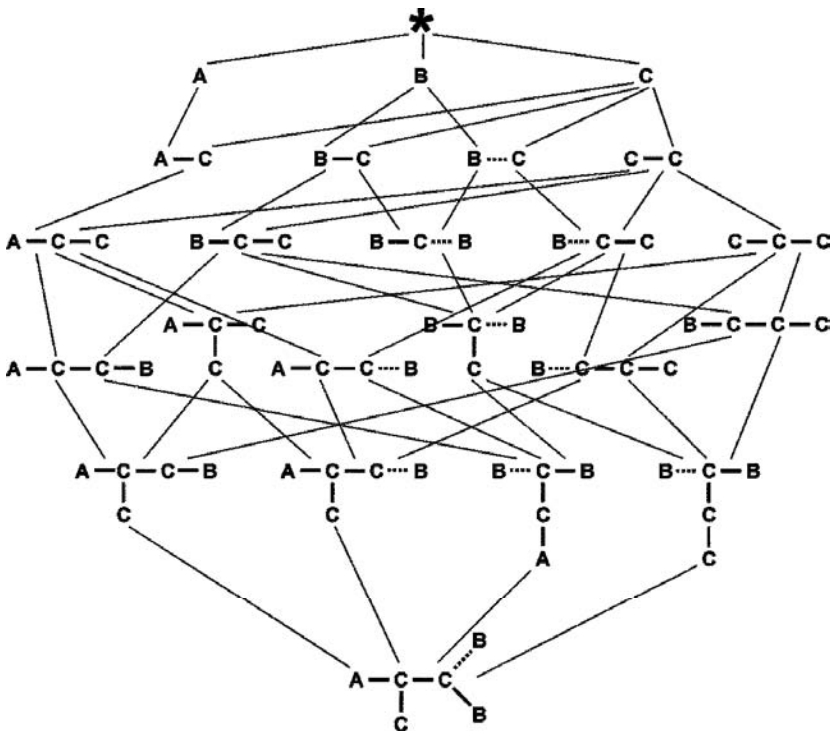
Another application area are web searches (Cook, Manocha, & Holder, 2003). Existing search engines use linear feature matches. Using graphs as underlying data structure, nodes represent pages, documents or document keywords and edges represent links between them. Posing a query as a graph means a smaller graph has to be embedded in the larger one. The graph modeling the data structure can be mined to find similar clusters.

Quite new is the application of subgraph mining in optimizing code for embedded devices. With the help of so-called procedural abstraction, the size of pre-compiled binaries can be reduced which is often crucial because of the limited storage capacities of embedded systems. There, subgraph mining helps identifying common structures in the program's control flow graph which can then be combined ("abstracted") into a single procedure (Dreweke et.al., 2007).

BACKGROUND

Theoretically, mining in graph databases can be modeled as the search in the lattice of all possible subgraphs. In Figure 1 a small example is shown based on one graph with six nodes labeled **A,B,C** as shown at the bottom of the figure. All possible subgraphs of this small graph are listed in this figure. At the top of the figure, the empty graph modeled with * is shown. In the next

Figure 1. The lattice of all subgraphs in a graph



row all possible subgraphs containing just one node (or zeros edges) are listed. The second row contains subgraphs with one edge. The "parent-child" relation between the subgraphs (indicated by lines) is the subgraph property. The empty graph can be embedded in every graph containing one node. The graph containing just one node labeled **A** can be embedded in a one edge graph containing nodes **A** and **C**. Please note, that in Figure 1 no graph with one edge is given containing nodes labeled **A** and **B**. As there is no such subgraph in our running example, the lattice does not contain a graph like this. Only graphs that are real subgraphs are listed in the lattice. In the third row, graphs with two edges are shown and so on. At the bottom of Figure 1, the complete graph with five edges is given. Each subgraph appearing in Figure 1 can be embedded in this graph. All graph mining algorithms have in common, that they search this subgraph lattice. They are interested in finding a subgraph (or several subgraphs) that can be embedded as often as possible in the graph to be mined. In Figure 1 the circled graph can be embedded twice in the running example.

When mining real life graph databases, the situation is of course much more complex. Not only one but a lot of graphs are analyzed leading to a very large lattice. Searching this lattice can be done depth or breadth first. When searching depth first in Figure 1, the first discovered subgraph will be **A** followed by **A-C**, **A-C-C** and so forth. Thus, first all subgraphs containing **A**, and in the next branch all containing **B** are found. If the lattice is traversed breadth first, all subgraphs in one level of the lattice, i.e. structures that have the same number of edges, are searched before the next level is started. The main disadvantage of breadth first search is the larger memory consumption because in the middle of the lattice a large amount of subgraphs has to be stored. With depth first search only structures which amount is proportional to the size of the biggest graph in the database have to be recorded during the search.

Building this lattice of frequent subgraphs involves two main steps: *Candidate Generation*, where new subgraphs are created out of smaller ones, and *Support Computation* where the frequency or support of the new subgraphs in the database is determined. Both steps are highly complex and thus various algorithms and techniques have been developed to find frequent subgraphs in finite time with reasonable resource consumptions.

MAIN THRUST OF THE CHAPTER

There are two popular ways of creating new subgraphs, new possible candidates for frequent fragments:

1. Merging smaller subgraphs that share a common core (Inokuchi et al., 2002; Kuramochi & Karypis, 2004), or
2. Extending subgraphs edge by edge (Borgelt & Berthold, 2002; Yan & Han, 2002).

The merge process can be explained by looking at the subgraph lattice shown in Figure 1. The circled subgraph has two parents, **A-C** and **C-C**. Both share the same core which is **C**. Thus the new fragment **A-C-C** is created by taking the core and adding the two additional edge-node pairs, one from each parent. There are two problems with this approach: First the common core needs to be detected somehow, which can be very expensive. Second a huge amount of subgraphs generated in this way may not even exist in the database. Merging e.g. **A-C** and **B-C** in the example will lead to **A-C-B** which does not occur in the database.

Extending fragments has the advantage that no cores have to be detected. New edge-node pairs (or sometimes only edges, if cycles are closed) are just added to an existing subgraph. In so called *embedding lists*, all embeddings of a fragment into the database graphs are stored. When extending a fragment with an edge (and a node probably), it is easy to check the current embeddings and find edges/nodes connected to these embeddings. This way only existing new fragments are generated. For example, in Figure 1, the embedding list of the circled fragment **A-C-C** contains two embeddings into the only database graph. Checking these embeddings and possible extensions in the database graph leads to new existing fragments.

The importance of a fragment depends on the number of times it appears in the database. If this number is given as a percentage of the number of database graphs, it is called *support*. If an absolute number is given, it is called *frequency*. Two ways of calculating the support/frequency are possible. First the graphs a fragment can be embedded in are counted, no matter how often it can be embedded in one graph. Second the embeddings itself are counted. In Figure 1 the circled fragment appears once in the database in the first case and twice in the second case. Counting embeddings can be done with subgraph isomorphism tests against all graphs in the database. These tests are NP-complete (Valiente, 2002). However there is a small improvement for this strategy, as it suffices to check for subgraph isomorphism only in the graphs where the parent graph(s) occur. Unfortunately this requires to keep a list of the graphs in which a subgraph occurs which can be quite memory consuming if the database is large.

The other way of calculating the support is by using the already introduced embeddings lists. An embedding can be thought of as a stored subgraph isomorphism i.e. a map from the nodes and edges in the subgraph to the corresponding nodes and edges in the graph. Now, if the support of a new extended subgraph has to be determined the position in the graph where it can occur is already known and only the additional node and edge have to be checked. This reduces the time to find the isomorphism but comes with the drawback of enormous memory requirements as all embeddings of a subgraph in the database have to be stored which can be millions for small subgraphs on even medium-sized databases of about 25,000 items. Using embedding lists the actual support for a structure can be determined by counting the number of different graphs that are referred to by the embeddings. This can be done in linear time.

Each possible fragment should only be created once, but it is obvious from the lattice in Figure 1, that there may exist several paths through the lattice to reach one fragment. These duplicates must be filtered out. For real life databases, it is also usually not possible to traverse the complete lattice because the number of subgraphs is too large to be handled efficiently. A mechanism is needed to prune the search tree that is built during the discovery process. If the support is calculated based on the number of graphs a fragment appears in, a supergraph of a graph that is infrequent must be infrequent, too. It cannot occur in more graphs than its parents in the lattice. This property is also known as the *antimonocity constraint*. Once the search reaches a point where a graph does not occur in enough items of the database any more this branch can be pruned. This leads to a drastic reduction of the number of subgraphs to be searched. When the support is calculated based on the embeddings, the antimonocity constraint holds for edge-disjoint embeddings. Therefore after each extension step, the biggest set of edge disjoint embeddings must be calculated. This problem equals the *Maximal Independent Set* problem for graphs, which is also NP-complete (Kuramochi & Karypis, 2005). An independent set is a set of vertices in a graph no two of which are adjacent.

To speed up the search even further various authors have proposed additional pruning strategies that shrink the search tree more efficiently while still finding all frequent fragments. Quite a few algorithms rely on so-called *canonical codes* that uniquely identify a subgraph. A subgraph's code is automatically built during the extension or merging process by e.g. recording the edges in the order as they were added to the graph. After each extension step, this code is compared to the canonical code for this subgraph, which is either the (lexicographically) smallest or biggest possible code. If the subgraph's code is not the canonical one, this branch in the search tree can be pruned, because the same subgraph will be found (or has already been found) in another branch (Borgelt, 2006).

FUTURE TRENDS

Despite the efforts of the last years, still several problems have to be solved. Memory and runtime are a challenge for most of the algorithms. Having real world graph databases containing millions of different graphs, various new algorithms and extensions of the existing ones are necessary. First thoughts concerning this topic can be found in (Wang, Wang, Pei, Zhu, & Shi, 2004). Another promising research direction are parallel and distributed algorithms. Distributing the graphs and their subgraph lattice onto different machines or using supercomputers with many processors and much memory can help in processing even larger databases than with current algorithms (Di Fatta & Berthold, 2006; Reinhard & Karypis, 2007).

In several application areas, it is not exact graph matching that is necessary. For example when mining molecules, it is helpful to search for groups of molecules having the same effect but not the same underlying graph. Well-known examples are the number of carbon atoms in chains or several carbon atoms in rings that have been replaced by nitrogen for example (Hofer, Borgelt & Berthold, 2003). In other areas, constraints must be imposed on the fragments to be found, not every possible fragment is helpful. Constraints can e.g. be the density of the graph or the minimal degree of nodes (Zhu, Yan, Han, & Yu, 2007).

Additionally visualization of the search and the results is difficult. A semi-automatic search can be helpful. A human expert decides whether the search in a subpart of the lattice is useful or if the search is more promising in another direction. To achieve this goal a visualization component is necessary that allows browsing in the graph database showing the embeddings of subgraphs.

Finally, another problem of most subgraph mining algorithms is the huge amount of frequent fragments they output. The user has then to scan them all and decide which ones are interesting for the specific application area. Such decision should be incorporated directly into the search process, making the result more understandable and often also the search faster.

CONCLUSION

Graph Data Mining is a currently very active research field. At the main data mining conferences of the ACM or the IEEE every year various new approaches appear. The application areas of graph data mining are widespread ranging from biology and chemistry to compiler construction. Wherever graphs are used to model data, data mining in graph data bases is useful.

REFERENCES

Borgelt, C., & Berthold, M. (2002, December). Mining Molecular Fragments: Finding Relevant Substructures of Molecules. *IEEE International Conference on Data Mining*, ICDM2002. Maebashi City, Japan, IEEE Press, 51-58.

Cook, D. J., & Holder L.B. (2000). Graph-Based Data Mining. *IEEE Intelligent Systems*, *15*(2), 32-41.

Cook, D.J., Manocha, N., & Holder, L.B. (2003). Using a Graph-Based Data Mining System to Perform Web Search. *International J. of Pattern Recognition and Artificial Intelligence*, *17*(5), 705-720.

Di Fatta, G., & Berthold, M.R. Dynamic Load Balancing for the Distributed Mining of Molecular Structures. *IEEE Transactions on Parallel and Distributed Systems, Special Issue on High Performance Computational Biology*, vol. 17, no. 8, IEEE Press, 773-785.

Dreweke, Alexander; Wörlein, Marc; Fischer, Ingrid; Schell, Dominic; Meinl, Thorsten; Philippsen, Michael (2007). Graph-Based Procedural Abstraction. *Proceedings of the Fifth International Symposium on Code Generation and Optimization*. IEEE Computer Society, 259-270.

Finn, P., Muggleton, S., Page D., & Srinivasan, A. (1998). Pharmacophore Discovery Using the Inductive Logic Programming System PROGOL. *Machine Learning, 30*(2-3), 241-270.

Helma, C., Kramer, S., & De Raedt, L. (2002, May). The Molecular Feature Miner MolFea. In: Hicks, M., & Kettner, C. (eds.) *Proceedings of the Beilstein-Institut Workshop Molecular Informatics: Confronting Complexity*. Bozen, Italy.

Hofer, H., Borgelt, C., & Berthold, M. (2003). Large Scale Mining of Molecular Fragments with Wildcards. In: Berthold, M., Lenz, H.J., Bradley, E., Kruse, R., & Borgelt, C. (eds.) *Advances in Intelligent Data Analysis V*, Lecture Notes in Computer Science 2810, Springer-Verlag, 380-389.

Huan J., Wang, W., & Prins, J. (2003). Efficient Mining of Frequent Subgraphs in the Presence of Isomorphism. *International Conference on Data Mining*, ICDM2003, Melbourne, Florida, USA, 549-552.

Inokuchi, A., Washio, T., & Motoda, H. (2000, September). An Apriori-based Algorithm for Mining Frequent Substructures from Graph Data. *4th European Conference on Principles of Data Mining and Knowledge Discovery*, PKDD2000, Lyon, France, 13-23.

Inokuchi, A., Washio, T., Nishimura, K., & Motoda, H. (2002). *A Fast Algorithm for Mining Frequent Connected Subgraphs*. IBM Research, Tokyo Research Laboratory.

Inokuchi, A., Washio, T., & Motoda, H. (2003). Complete Mining of Frequent Patterns from Graphs: Mining Graph Data. *Machine Learning, 50*(3), 321-354.

King, R., Srinivasan, A., & Dehaspe, L. (2001). Warmr: A Data Mining Tool for Chemical Data. *Journal of Computer-Aided Molecular Design*, 15, 173-181.

Kuramochi M., & Karypis G. (2004, September) An Efficient Algorithm for Discovering Frequent Subgraphs. In: *IEEE Transactions on Knowledge and Data Engineering*, 16(9), 1038-1051, Piscataway, NJ, USA.

Kuramochi, M., & Karypis G (2005). Finding Frequent Patterns in a Large Sparse Graphs. *Data Mining and Knowledge Discovery, 11*(3), 243-271.

Nijssen, S., & Kok, J. (2004, April) The Gaston Tool for Frequent Subgraph Mining. *Electronic Notes in Theoretical Computer Science, 127*(1), Elsevier, 77-87.

Reinhard, S., & Karypis, G. (2007, March). A Multi-Level Parallel Implementation of a Program for Finding Frequent Patterns in a Large Sparse Graph. *12th International Workshop on High-Level Parallel Programming Models and Supportive Environments*. Long Beach, CA, USA.

Rückert, U., & Kramer, S. (2004, March). Frequent Free Tree Discovery in Graph Data. In Haddad, H., Omicini, A., Wainwright R., & Liebrock, L. (Eds.), *ACM Symposium on Applied Computing*, SAC 2004, Nicosia, Cyprus, 564-570.

Valiente, G. (2002). *Algorithms on Trees and Graphs*. Springer-Verlag.

Wang, C., Wang, W., Pei, P., Zhu, Y., & Shi, B. (2004, August). Scalable Mining Large Disk-based Graph Databases. *10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD 2004, Seattle, WA, USA, 316-325.

Yan, X., & Han, J. (2002, December). gSpan: Graph-Based Substructure Pattern Mining. *IEEE International Conference on Data Mining*, ICDM 2002, Maebashi City, Japan, 721-724.

Zhu, F., Yan, X., Han, J., & Yu, P. (2007, May). gPrune: A Constraint Pushing Framework for Graph Pattern Mining, *Pacific-Asia Conf. on Knowledge Discovery and Data Mining*, Nanjing, China.

KEY TERMS

Antimonicity Constraint: The antimonicity constraint states, that any supergraph of an infrequent graph must be infrequent itself.

Candidate Generation: Creating new subgraphs out of smaller ones; then it is checked how often this new subgraph appears in the analyzed graph data base.

Canonical Code: A unique linear representation of a graph that can be used to prune the search tree

Frequent Subgraph: a subgraph that occurs in a certain percentage of all graphs in the database.

Graph isomorphism: Two graphs which contain the same number of graph vertices connected in the same way by edges are said to be isomorphic. Determining if two graphs are isomorphic is thought to be neither

an NP-complete problem nor a P-problem, although this has not been proved (Valiente, 2000).

Search Tree Pruning: Cutting of certain branches of the (conceptual) search tree that is built during the mining process; pruning criteria may be the size of the graphs, the support of the graphs, or algorithm-specific constraints.

Subgraph: A graph G' whose vertices and edges form subsets of the vertices and edges of a given graph G. If G' is a subgraph of G, then G is said to be a supergraph of G'.

Subgraph Isomorphism: Decision whether a graph *G'* of is isomorphic to a subgraph another graph *G*. This problem is known to be NP-complete.

Support: The number of graphs or embeddings in the analysed database in which a subgraph occurs.

Subsequence Time Series Clustering

Jason Chen
Australian National University, Australia

INTRODUCTION

Clustering analysis is a tool used widely in the Data Mining community and beyond (Everitt et al. 2001). In essence, the method allows us to "summarise" the information in a large data set X by creating a very much smaller set C of representative points (called centroids) and a membership map relating each point in X to its representative in C. An obvious but special type of data set that one might want to cluster is a time series data set. Such data has a temporal ordering on its elements, in contrast to non-time series data sets. In this article we explore the area of time series clustering, focusing mainly on a surprising recent result showing that the traditional method for time series clustering is meaningless. We then survey the literature of recent papers and go on to argue how time series clustering can be made meaningful.

BACKGROUND

A time series is a set of data points which have temporal order. That is,

$$X = \{x_t \mid t = 1, ..., n\} \quad (1)$$

where t reflects the temporal order. Two types of clustering of time series has historically been undertaken: whole series clustering and subsequence clustering. In whole series clustering, one generally has a number of time series of equal length (say n) and one forms a vector space of dimension n so that each time series is represented by a single point in the space. Clustering then takes place in the usual way and groupings of similar time series are returned.

Whole series clustering is useful in some circumstances, however, often one has a single long time series data set X and the aim is to find a summary set of features in that time series, e.g. in order to find repeating features or particular repeating sequences of features (e.g. see the rule finding method proposed in (Das et al.1998)). In this case, what was historically done was to create a set Z of subsequences by moving a sliding window over the data in X, i.e.

$$z_{p-(w-1)} = x_{p-(w-1)}, x_{p-(w-2)}, ..., x_{p-2}, x_{p-1}, x_p \quad (2)$$

$z_p \in Z$, $p = w...n$. Each subsequence z_p (also called more generally a regressor or delay vector; see below) essentially represents a feature in the time series. These features live in a w-dimensional vector space, and clustering to produce a summarising set C of "centroid" features can proceed in the usual way. This technique has historically been called Subsequence Time Series (STS) Clustering, and quite a lot of work using the technique was published (see (Keogh et al. 2003) for a review of some of this literature). In this article we will focus on the area of subsequence time series clustering. For a review of whole time series clustering methods, see (Wang et al. 2004).

Given the widespread use of STS clustering, a surprising result in (Keogh et al. 2003) was that it is meaningless. Work in (Keogh et al. 2003) defined a technique as meaningless if the result it produced was essentially independent of the input. The conclusion that STS clustering was meaningless followed after it was shown that, if one conducted STS clustering on a range of even very distinct time series data sets, then the cluster centroids resulting from each could not be told apart. More specifically, the work clustered each time series multiple times and measured the average "distance" (see (Keogh et al. 2003) for details) between clustering outcomes from the same time series and between different time series. They found on average that the distance between clustering outcomes from the same and different time series were the same. Further, they discovered the strange phenomenon that the centroids produced by STS clustering are smoothed sine-type waves.

After the appearance of this surprising result, there was great interest in finding the cause of the dilemma and a number of papers on the topic subsequently appeared. For example, Struzik (Struzik 2003) proposed that the

"meaningless" outcome results only in pathological cases, i.e. when the time series structure is fractal, or when the redundancy of subsequence sampling causes trivial matches to hide the underlying rules in the series. They suggested autocorrelation operations to suppress the latter, however these suggestions were not confirmed with experiments.

In contrast, Denton (Denton, 2005) proposed density based clustering, as opposed to, for example, k-means or hierarchical clustering, as a solution. They proposed that time series can contain significant noise, and that density based clustering identifies and removes this noise by only considering clusters rising above a preset threshold in the density landscape. However, it is not clear whether noise (or only noise) in the time series is the cause of the troubling results in (Keogh et al. 2003). For example, if one takes the benchmark Cylinder-Bell-Funnel time series data set (see (Keogh et al. 2003)) without noise and applies STS clustering, the strange smoothed centroid results first identified there are still returned.

Another interesting approach to explain the dilemma was proposed by Goldin et. al. (Goldin et al. 2006). They confirmed that the ways (multiple approaches were tried) in which distance between clustering outcomes were measured in (Keogh et al. 2003) did lead to the conclusion that STS-clustering was meaningless. However, they proposed an alternative distance measure which captured the "shape" formed by the centroids in the clustering outcome. They showed that if one calculates the average shape of a cluster outcome over multiple clustering runs on a time series, then the shape obtained can be quite specific to that time series. Indeed if one records all the individual shapes from these runs (rather than recording the average), then in an experiment on a set of ten time series they conducted, one is able to match a new clustering of a time series back to one of the recorded clustering outcomes from the same time series. While these results suggest meaningfulness is possible in STS-clustering, it seems strange that such lengths are required to distinguish between clustering outcomes of what can be very distinct time series. Indeed we will see later that an alternative approach, motivated from the Dynamical Systems literature, allows one to easily distinguish between the clustering outcomes of different time series using the simple distance measure adopted in (Keogh et al. 2003).

Another approach proposed by Chen (Chen 2005, 2007a) to solve the dilemma forms the basis of work which we later argue provides its solution. They proposed that the metrics adopted in (Keogh et al. 2003) in the clustering phase of STS clustering were not appropriate and proposed an alternative clustering metric based on temporal and formal distances (see (Chen 2007a) for details). They found that meaningful time series clustering could be achieved using this metric, however the work was limited in the type of time series to which it could be applied. This work can be viewed as restricting the clustering process to the subset of the clustering space that was visited by the time series; a key tenet of later work that we argue below forms a solution to the STS-clustering dilemma.

Peker (Peker 2005) also conducted experiments in STS-clustering of time series. They identified that clustering with a very large number of clusters leads to cluster centroids that are more representative of the signal in the original time series. They proposed the idea of taking cluster cores (a small number of points in the cluster closest to the centroid) as the final clusters from STS clustering. The findings in this work concur with work in (Chen 2007a) and the work we explore below, since they are compatible with the idea of restricting clustering to the subset of the clustering space visited by the time series.

While each of the works just reviewed show interesting results which shed light on the problems involved with STS-clustering, none provides a clear demonstration for general time series of how to overcome them.

MAIN FOCUS

We now propose our perspective on what the problem with STS clustering is, and on a solution to this problem; based on a number of recent papers in the literature. Let us revisit the problems found in (Keogh et al. 2003) with the STS clustering method. This work proposed that STS-clustering was meaningless because:

A. One could not distinguish between the clustering outcomes of distinct time series, even when the time series themselves were very different, and
B. Cluster representatives were smoothed and generally did not look at all like any part of the original time series

They proposed that these two problems were one and the same, i.e. that one could not distinguish between cluster centres of different time series because they were all smoothed, and hence alike. This presumption turns out to be false, i.e. really these are two separate problems which need to be addressed and solved separately. For example, (Chen 2007b) showed how a time series clustering technique could produced distinguishable cluster centres (i.e. overcome (A)) but still produce centres that were smoothed (i.e. not overcome (B)). Hence, (Chen 2007b) proposed a new set of terminologies to reflect this fact. They proposed that a time series clustering method which overcomes problem (A) should be called meaningful, and one that overcomes problem (A) and (B) should be called useful. We will expand later on the motivation behind why these terms were adopted in each case.

Recall how it was shown in (Keogh et al. 2003) that STS-clustering is meaningless; the work clustered each time series multiple times and then measured the distance between clustering outcomes from the same time series and between different time series. Work in both (Chen 2007b) and (Simon et al. 2006) proposed that what was required to make the STS-clustering method meaningful was to introduce a lag q into the window forming process. That is, form subsequences as,

$$z_{p-(w-1)q} = x_{p-(w-1)q}, x_{p-(w-2)q}, ..., x_{p-2q}, x_{p-1q}, x_p \qquad (3)$$

$z_p \in Z$, $p = (w-1)q+1...n$ (i.e. so that now adjacent points in the subsequence are separated by q data points in the time series) where we call z_p a regressor or delay vector. The inspiration of both works was from the field of Dynamical Systems (Sauer et al. 1991, Ott et al. 1994) where it is well known that introducing a lag is required for the embedding of any real world (i.e. noisy and represented with limited precision) time series in a vector space using a sliding windows type process. Geometrically, not using a lag means subsequence vectors will be clumped along the diagonal of the space, and hence, even with a small amount of noise present in the time series, and reasonable precision, the "information" in the embedding that distinguishes one time series from another is lost. Work in (Simon et al. 2006) went on to conduct the same experiment as in (Keogh et al. 2003) (albeit with different time series), but using a lag, and found that cluster centres produced from distinct time series were then indeed distinguishable. Work in (Chen 2007b) confirmed the result in (Simon et al. 2006) using basically the same time series as used in (Keogh et al. 2003). For clarity, and to distinguish between what follows, we follow the terminology adopted in (Chen 2007b) and denote the STS-clustering technique where a lag is introduced into the sliding windows process as Unfolded Time Series (UTS) clustering.

According to the "meaningful" and "useful" terminology introduced above, the UTS clustering method produces meaningful clustering outcomes. That is, if we cluster two distinct time series using the method, then UTS clustering produces centroid sets in each case which are distinct from one another. In essence, the "information" existing in the original time series which made them distinct has been retained in the clustering outcome, and so the clustering outcome really can be described as meaningful. Hence, the problem of achieving meaningful time series clustering would seem solved, i.e. one must introduce a lag into the subsequence vector construction process.

This could mark the end of the dilemma. However, recall the second problem ((B) above) observed by Keogh with STS clustering; that centroids are smoothed and do not look like, or retain the properties of, the original time series. Work in (Chen 2007b) noted that the UTS-clustering method, although meaningful, was still prone to this second problem, i.e. according to our adopted terminology it is not a useful time series clustering method. The term "useful" was adopted in (Chen 2007b) based on the observation that one clusters a time series to produce a summary set of features in the time series. If these features do not look like any part of the time series, then the outcome, although meaningful, is not useful. Why should UTS clustering be meaningful, but not produce centroids representative of the time series?

To answer this question, (Chen 2007b) proposed that we need to look more fundamentally at what we are asking when we UTS (or STS) cluster a time series. If we UTS cluster with a sliding window length of d, then we form a d dimensional clustering space $\Re^d$. In its entirety, $\Re^d$ represents the full range of possible subsequence (i.e. feature) shapes and magnitudes that can exist. However, (Chen 2007b) noted that the underlying system producing the time series almost certainly will not live on all of $\Re^d$, or indeed even on a convex subset of $\Re^d$ (something assumed by typical clustering algorithms like k-means and Expectation Maximization used in STS clustering to date). What

sense does it make to include in the clustering process parts of $\Re^d$ that cannot be realised in the underlying system? Work proposing methods for clustering on subspaces (Haralick & Harpaz 2005), and manifolds (Breitenbach & Grundic 2005) exists and is motivated by exactly this line of thinking. Some simple experiments were conducted in (Chen 2007b) to show that this unrestricted approach to clustering in UTS (STS) clustering is the root cause of the smoothed centroid problem.

So we should cluster only in the subset of $\Re^d$ where valid outcomes from the underlying system exist. Unfortunately, given only a finite time series produced by the system, one cannot know the extent of this subset. However, this need not matter if the aim of clustering a time series is to (a) summarise the time series that was seen, rather than (b) to summarise the possible time series outcomes of an underlying system. (Chen 2007b) proposed that (a) is generally what we want to do when clustering a time series, and corresponds to asking the question: given the features observed in a time series, which k (for k clusters) of these features best "summarises" the time series. Given this observation, (Chen 2007b) went on to propose a method that restricts the clustering process to the region in $\Re^d$ visited by the time series. They proposed that this approach corresponds to the correct way to apply the clustering technique if indeed we want to ask the question corresponding to (a). They called the approach the Temporal-Formal (TF) clustering algorithm.

Details of the results of applying the technique can be found in (Chen 2007b), however in summary, the technique was applied on key time series data sets adopted from (Keogh et al. 2003) and (Simon et al. 2006). The results for all time series in the data set were,

1. Centroids were produced which remained in among data points in the cluster they represented, i.e. centroids looked like features from the original time series
2. Clustering outcomes were meaningful (as per the definition of meaningful above) in all cases.

The conclusion was therefore made that the TF clustering algorithm was a useful time series clustering method. Further, analysis of the clustering outcomes for a number of time series was conducted in (Chen 2007b), including the benchmark Cylinder-Bell-Funnel time series. In each case the TF-clustering algorithm lead to the intuitively correct or (in the case of the benchmark time series) required outcome.

FUTURE TRENDS

Subsequence clustering of time series has been the focus of much work in the literature, and often as a subroutine to higher level motivations such as rule discovery, anomaly detection, prediction, classification and indexing (see (Keogh et al. 2003) for details). With the discovery that STS-clustering is meaningless, much future work will involve revisiting and reviewing the results and conclusions made by this work. Of great importance then is the discovery of a meaningful subsequence time series clustering method. We have argued here that a means for the solution of the problem exists, and while the arguments made in this work seem both clear and cogent, this work is quite recent. The dust still has not settled in the time series clustering area of data mining research.

CONCLUSION

We have reviewed the area of time series clustering, focusing on recent developments in subsequence time series (STS) clustering. Prior to 2003, STS clustering was a widely accepted technique in data mining. With the discovery in (Keogh et al. 2003) that STS-clustering is meaningless, a number of articles were published to explain the dilemma. While interesting results were presented in all these papers, we argued that two papers provide a solution to the dilemma ((Simon et al. 2006) and (Chen 2007b)). Specifically, work in (Keogh et al. 2003) identified two problems with STS clustering: (A) and (B) as described above. Together the papers show that these problems can be solved (respectively) by (a) introducing a lag into the sliding windows part of the STS-clustering process, and (b) restricting the clustering process to only that part of the clustering space pertinent to the time series at hand. Hence, we propose that this work forms a solution to the STS-clustering dilemma first identified in (Keogh et al., 2003).

REFERENCES

Denton, A. (2005). Kernel-density-based clustering of time series subsequences using a continuous random-walk noise model, in *Proceedings of IEEE International Conference on Data Mining*, Houston, USA.

Everitt, B.S., Landau,S., & Leese, M. (2001). *Clustering Analysis*, Wiley.

Goldin,D., Mardales, R., & Nagy, G. (2006). In search of meaning for time series subsequence clustering: matching algorithms based on a new distance measure, in *Proceedings of Conference of Information and Knowledge Management*, Arlington, USA.

Keogh, E., Lin, J., & Truppel, W. (2003). Clustering of time series subsequences is meaningless: Implications for previous and future research, in *Proceedings of the International Conference of Data Mining*.

Ott, E., Sauer, T., & Yorke, J. (1994). *Coping with Chaos*, Wiley.

Das, G., Lin, K., Mannila, H., Renganathan, G., & Smyth, P. (1998). Rule discovery from time series, in *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining*, New York, NY.

Simon, G., Lee, J.A., & Verleysen, M. (2006). Unfolding preprocessing for meaningful time series clustering, *Neural Networks* 19, 877–888.

Chen, J.R. (2005). Making subsequence time series clustering meaningful, in *Proceedings of IEEE International Conference on Data Mining*, Houston, USA, pp. 114–121.

Chen, J.R. (2007a). Making clustering in delay vector space meaningful, *Knowledge and Information Systems* ,11(3), 369–385.

Chen, J.R. (2007b). Useful clustering outcomes from meaningful time series clustering, in *Proceedings of the Australasian Data Mining Conference*, Gold Coast, Australia.

Peker, K. (2005). Subsequence time series (sts) clustering techniques for meaningful pattern discovery, in *International Conference Integration of Knowledge Intensive Multi-Agent Systems (KIMAS)*,Waltham, USA.

Breitenbach, M., & Grundic, G.Z. (2005). Clustering through ranking on manifolds, in *Proceedings of the 22nd International Conference on Machine Learning*, Bonn, Germany.

Haralick, R.M., & Harpaz, R. (2005). Linear manifold clustering, in *Proceedings of the International Conference on Machine Learning and Data Mining*, Leipzig.

Sauer, T., Yorke, J., & Casdagli, M. (1991). Embedology, *Journal of Statistical Physics* 65, 579.

Wang, X., Smith, K.A., Hyndman, R., & Alahakoon, D. (2004). A scalable method for time series clustering, *Technical report of Department of Econometrics and Business Statistics*, Monash University, Victoria, Australia.

Struzik, Z.R. (2003). Time Series Rule Discovery: Tough, Not Meaningless, *Foundations of Intelligent Systems, Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 32–39.

KEY TERMS

Delay Vector: Elements of the data set obtained in UTS clustering, i.e. by using Equation 3 above.

Lag: The sliding window used in UTS clustering need not capture, as a delay vector, a sequence of adjacent points in the time series. The lag is the value $q = p+1$ where p is the number of data points in the time series lying between adjacent points in the delay vector. So, for example, a lag $q = 3$ means the first delay vector will be x_1, x_4, x_7, ...

Meaningless: in general, an algorithm is said to be meaningless if its output is independent of its input. In the context of time series clustering, a time series clustering algorithm is said to be meaningless if one cannot distinguish between the clustering outcomes of distinct time series.

Regressor: Elements of the data set obtained in UTS clustering, i.e. by using Equation 3 above.

Subsequence Time Series (STS) Clustering: The process of applying standard clustering techniques to a dataset whose elements are constructed by passing a sliding window over (usually) a single (long) time series.

Subsequence Vector: Elements of the data set obtained in STS clustering, i.e. by using Equation 2 above.

Temporal-Formal (TF) Clustering: UTS clustering where the clustering process is restricted to the region in the clustering space that was visited by the time series.

Time Series: A data set containing elements which have a temporal ordering

Unfolded Time Series (UTS) Clustering: UTS clustering is STS clustering where a lag greater than unity has been introduced into the sliding windows process.

Whole Time Series Clustering: The process of applying standard clustering techniques to a dataset whose elements are distinct time series of equal length.

Summarization in Pattern Mining

Mohammad Al Hasan
Rensselaer Polytechnic Institute, USA

INTRODUCTION

The research on mining interesting patterns from transactions or scientific datasets has matured over the last two decades. At present, numerous algorithms exist to mine patterns of variable complexities, such as set, sequence, tree, graph, etc. Collectively, they are referred as **Frequent Pattern Mining** (FPM) algorithms. FPM is useful in most of the prominent knowledge discovery tasks, like classification, clustering, outlier detection, etc. They can be further used, in database tasks, like indexing and hashing while storing a large collection of patterns. But, the usage of FPM in real-life knowledge discovery systems is considerably low in comparison to their potential. The prime reason is the lack of interpretability caused from the enormity of the output-set size. For instance, a moderate size graph dataset with merely thousand graphs can produce millions of frequent graph patterns with a reasonable support value. This is expected due to the combinatorial search space of pattern mining. However, classification, clustering, and other similar Knowledge discovery tasks should not use that many patterns as their knowledge nuggets (features), as it would increase the time and memory complexity of the system. Moreover, it can cause a deterioration of the task quality because of the popular "*curse of dimensionality*" effect. So, in recent years, researchers felt the need to summarize the output set of FPM algorithms, so that the summary-set is *small*, *non-redundant* and *discriminative*. There are different summarization techniques: lossless, profile-based, cluster-based, statistical, etc. In this article, we like to overview the main concept of these summarization techniques, with a comparative discussion of their strength, weakness, applicability and computation cost.

BACKGROUND

FPM had been the core research topic in the field of data mining for the last decade. Since, its inception with the seminal paper of mining association rules by Argrawal *et al* (Agrawal & Srikant, 1994), it has matured enormously. Currently, we have very efficient algorithms for mining patterns with higher complexity, like sequence (Zaki, 2001), tree (Zaki, 2005) and graph (Yan & Han, 2002; Hasan, Chaoji, Salem & Zaki, 2005). The objective of FPM is as follows: Given a database D, of a collection of events (an event can be as simple as a set or as complex as a graph) and a user defined support threshold π^{min}; return all patterns (patterns can be set, tree, graph, etc. depending on D) that are frequent with respect to π^{min}. Sometimes, additional constraints can be imposed besides the minimum support criteria. For details on FPM, see data mining textbooks (Han & Kamber, 2001).

FPM algorithms search for patterns in a combinatorial search space, which is generally very large. But, the ***anti-monotone*** property allows fast pruning: which states, *"If a pattern is frequent, so is all it's sub-pattern; if a pattern is infrequent, so is all its super-pattern."* Efficient data structure and algorithmic techniques on top of this basic principle enable FPM algorithms to work efficiently on database of millions events. However, the usage of frequent patterns in knowledge discovery tasks requires the analysts to set a reasonable support value for the mining algorithms to obtain interesting patterns, which, unfortunately, is not that straightforward. Experience suggests that if the support value is set too high, only few common-sense patterns are obtained. For example, if the database events are recent movies watched by different subscribers, using a very high support will only return the set of super-hit movies which are liked by anybody. On the other hand, setting low support value returns enormously large number of frequent patterns that are difficult to interpret; many of those are redundant too. So, ideally one would like to set the support value at a comparably lower threshold and then adopt a summarization or compression technique to obtain a smaller *FP*-set, comprising interesting, non-redundant, and representative patterns.

Figure 1: An itemset database of 6 transactions (left). Frequent, Maximal and Closed patterns mined from the dataset in 50% support (right)

Transaction Database, D
1. A C T W 2. C D W 3. ACTW 4. ACDW 5. ACDTW 6. CDT

Frequent Patterns in D (Minimum Support = 3)			
Support	**Frequent Pattern**	**Maximal Pattern**	**Closed Pattern**
6	C		C
5	W, CW		CW
4	A, D, T, AC, AW, CD, CT, ACW		CD, CT, ACW
3	AT, DW, TW, ACT, ATW, CDW, CTW, ACTW	CDW, ACTW	CDW, ACTW

MAIN FOCUS

The earliest attempt to compress the FPM result-set was to mine Maximal Patterns (Bayardo, 1998). A frequent pattern is called maximal, if it is not a sub-pattern of any other frequent pattern (see the example in figure 1). Depending on the dataset, maximal pattern can reduce the result-set substantially; especially for dense dataset, the compression ratio can be very high. And, maximal patterns can be mined in the algorithmic framework of FPM processes without a post-processing step. The limitation of maximal pattern mining is that the compression also loses the support information of the non-maximal patterns; for example, in figure 1, from the list of maximal patterns we can deduce that the pattern *CD* is also frequent (since *CDW* is frequent), but its support value is lost (which is 4, instead of 3). Support information is critical if the patterns are used for rule generation, as it is essential for confidence computation. To circumvent that, Closed Frequent Pattern Mining was proposed by Zaki (Zaki, 2000). A pattern is called closed, if it has no super-pattern with the same support. The compressibility of closed frequent mining is smaller than the maximal pattern mining, but for the earlier, all frequent patterns and also, their support information can be immediately retrieved (without further scan of the database). Closed frequent pattern can also be mined within the FPM process.

Pattern compression offered by maximal or closed mining framework is not sufficient, as the result-set size is still too large for human interpretation. So, many pattern summarization techniques have been proposed lately, each with different objective preference. It is difficult and sometimes, not fair, to compare them. For instance, in some cases, the algorithms try to preserve the support value of the frequent patterns; whereas, in other cases the support value is completely ignored and more emphasis is given in controlling the redundancy in patterns. In the following few paragraphs we discuss the main ideas of some of the major compression approaches, with their benefits and limitations. At the end of this section (see table 1), we show the benefits/limitations of these algorithms in tabular form for quick references.

Top-*k* Patterns

If the analyst has a predefined number of patterns in mind that (s)he wants to employ in the knowledge discovery tasks, *top-k patterns* is one of the best summarization technique. Han et al. (Han, Wang, Lu & TzVetkov, 2002) proposed one of the earliest algorithms that falls in this category. Their top-*k* patterns are *k* most frequent closed patterns with a user-specified minimum-length, *min_l*. Note that, minimum-length constraint is essential, since without it only length-1 patterns (or their corresponding closed super-pattern) will be reported, since they always have the highest frequency. The authors proposed efficient implementation of their proposed algorithm using FP-Tree; un-

fortunately, this framework works on itemset patterns only. Nevertheless, the summarization approach can be extended to other kind of patterns as well. Since, support is a criterion for choosing summary patterns; this technique is a support-aware summarization.

Another top-*k* summarization algorithm is proposed by Afrati et al. (Afrati, Gionis & Mannila, 2004). Pattern support is not a summarization criterion for this algorithm, rather high compressibility with maximal coverage is its primary goal. To achieve this, it reports maximal patterns and also, allows some false positive in the summary pattern set. A simple example from their paper is as follows: if the frequent patterns containing the sets *ABC, ABD, ACD, AE, BE,* and all their subsets, a specific setting of their algorithm may report *ABCD* and *ABE* as the summarized frequent patterns. Note that, this covers all the original sets, and there are only two false positive (*BCD, ABCD*). Afrati *et al.* showed that, the problem of finding the best approximation of a frequent itemsets that can be spanned by *k* set is NP-Hard. They also provided a greedy algorithm that guarantees an approximation ratio of at least $(1 - 1/e)$. The beauty of this algorithm is its high compressibility; authors reported that by using only 20 sets (7.5% of the maximal patterns), they could cover 70% of the total frequent set collection with only 10% false-positive. Also note, in such a high compression, there won't be much redundancy in the reported patterns; they, indeed, will be very different from each other. The drawbacks are: firstly, it is a post-processing algorithm, *i.e.*, all maximal patterns first need to be obtained to be used as an input to this algorithm. Secondly, the algorithm will not generalize well for complex patterns, like tree or graph, as the number of false-positives will be much higher and the coverage will be very low. So the approximation ratio mentioned above will not be achieved. Thirdly, it will not work if the application scenario does not allow false-positive.

Xin *et al.* (Xin, Cheng, Yan & Han, 2006) proposed another top-*k* algorithm. To be most effective, the algorithms strive for the set of patterns that collectively offer the best significance with minimal redundancy. Significance of a pattern is measured by a real value that encodes the degree of interestingness or usefulness of it and redundancy between a pair of patterns is measured by incorporating pattern similarity. Xin and his colleagues showed that to find such a set of top-*k* patterns is NP-Hard even for itemset, which is the simplest kind of pattern. They also proposed a greedy algorithm that approximates the optimal solution with $O(\ln k)$ performance bound. The major drawback of this approach is that the users need to find all the frequent patterns and evaluate their significances before the process of finding top-*k* patterns is initiated. Again, for itemset pattern the distance calculation is very straightforward, this might be very costly for complex patterns.

Support-Preserving Pattern Compression

We already discuss "Closed pattern mining", which is one approach of support preserving pattern compression. However, there are other support-preserving pattern mining algorithms that apply to itemset only. One of the most elegant among these is ***Non-Derivable Frequent Itemsets*** (Calders & Goethals, 2007). The main idea is based on *inclusion-exclusion principle*, which enables us to find a lower bound and upper bound on the support of an itemset based on the support of its subsets. These bounds can be represented by a set of derivable rules, which can be used to derive the support of the super itemset from its sub itemset. If the support can be derived exactly, the itemset are called *derivable itemsets* and need not be listed explicitly. For example, for the database in Figure 1, the itemset *CTW* is derivable, because the following two rules hold:

$$\sup(CTW) \geq \sup(TC) + \sup(TW) - \sup(T) \text{ and}$$
$$\sup(CTW) \leq \sup(TW).$$

From these, the support of CTW can be deduced exactly to be 3. Thus, a concise representation of frequent itemset consists of the collection of only *non-derivable itemsets*. Of course, the main limitation of this compression approach is that it applies only for itemset patterns; since, the inclusion-exclusion principle does not generalize for patterns with higher order structures. The compressibility of this approach is not that good either. In fact, Calders and Goethals proved that for some datasets, number of closed frequent patterns can be smaller in size than that of non-derivable frequent patterns. Moreover, there is no guarantee regarding redundancy in output set considering many of the frequent patterns are very similar to each other.

Cluster-Based Representative Pattern-Set

Pattern compression problem can be perceived as a clustering problem, if a suitable distance function between frequent patterns can be adopted. Then the summarization task is to obtain a set of representative patterns; each elements of the set resembles a cluster centroid. As typical clustering problem, this approach also leads to a formulation which is NP-Hard. So, sub-optimal solutions with known bounds are sought. Xin et al. (Xin, Han, Yan & Cheng, 2005) proposed greedy algorithms, named **RPglobal** and **RPlocal** to summarize itemset patterns. The distance function that they used considers both pattern object and its support. The representative element, P_r of a cluster containing elements, P_1, P_2… P_k, satisfies

$$\bigcup_{i=1}^{k} P_i \subseteq P_r.$$

And the distance is computed in transaction space. For instance, if two patterns P_1 and P_2 occur in transaction {1, 3, 5} and {1, 4, 5}, respectively, distance between them can be computed as,

$$1 - \frac{|\{1,3,5\} \cap \{1,4,5\}|}{|\{1,3,5\} \cup \{1,4,5\}|} = 0.5.$$

To realize a pattern cluster, a user defined $\boldsymbol{\delta}$ is chosen as distance threshold. If a pattern has a distance less than $\boldsymbol{\delta}$ from the representative, the pattern is said to be covered by the representative. **RPglobal** algorithm greedily selects the representative based on the remaining patterns to be covered. **RPlocal** finds the best cover set of a pattern by a sequential scan of the output set. For both the algorithms, the authors provided a bound with respect to the optimal number of representative patterns. For details, see the original paper (Xin, Han, Yan & Cheng, 2005). The attraction of cluster-based representative is that the algorithm is general and can easily be extended to different kinds of patterns. We just need to define a distance metric for that kind of pattern. The drawback is that user need to choose the right value of $\boldsymbol{\delta}$, which is not obvious.

Profile-Based Pattern Summarization

Yan *et al.* (Yan, Cheng, Han, & Xin, 2005) proposed profile-based pattern summarization, again, for itemset patterns. A profile-based summarization finds *k* representative patterns (named as Master patterns by Yan *et al*.) and builds a generative model under which the support of the remaining patterns can be easily recovered. A Master pattern is the union of a set of very similar patterns. To cover the span of the whole frequent patterns, a Master pattern is very different from another Master pattern. Note that, similarity considers both the pattern space and their support, so that they can be representative in the sense of FPM. The authors also built a profile for each Master pattern. Using the profile, the support of each frequent pattern that the corresponding Master pattern represents can be approximated without consulting the original dataset. The definition of Pattern profile is as follows: Firstly, for any itemset α, let D_α be the transaction-set that contains α; D be the entire dataset, and α_1, α_2,..., α_l be a set of similar patterns. Now, define

$$\mathsf{D}' = \bigcup_{i=1}^{l} \mathsf{D}_{\alpha_i}.$$

A profile M over α_1, α_2,..., α_l is a triple

$$\langle \mathsf{P}, \phi, \rho \rangle,$$

where, P is a probability distribution vector of the items α_1, α_2,..., α_l learned from the set D',

$$\varnothing = \bigcup_{i=1}^{l} \alpha_i$$

is taken as the Master pattern of α_1, α_2,..., α_l and

$$\rho = \frac{|\mathsf{D}'|}{|\mathsf{D}|}$$

is regarded as the support of the profile. For example, if itemset CT (D_{CT}={1,3,5,6}) and CW (D_{CW}={1,2,3.4,5}) of table 1 are to be merged in a profile, the Master pattern is CTW, its support is

$$\frac{|D'|}{|D|} = \frac{6}{6} = 100\%$$

(note that, original pattern CTW has a support of 83%) and $P = (4/6, 5/6)$. Profile based summarization is elegant due to its use of probabilistic generative model, but there are drawbacks. Authors did not proof any bound on the quality of results, so depending on the dataset, the summarization quality can vary arbitrarily. It is a post-processing algorithm, so all frequent patterns need to be obtained first. Finally, clustering is a sub-task of their algorithm; since, clustering itself is an NP-Hard problem, any solution adopting clustering algorithms, like k-means, hierarchical and etc. can only generate a local optimal solution.

Very recently, Wang and Parthasarathy (Wang & Parthasarathy, 2006) used probabilistic profile to summarize frequent itemsets. They used Markov Random Field (MRF) to model the underlying distribution that generates the frequent patterns. To make the model more concise, the authors considered only the non-derivable frequent itemsets instead of the whole set. Once the model is built, the list of frequent itemsets and associated count can be recovered using standard probabilistic inference methods. Probabilistic methods provide the most compact representation with very rough support estimation.

FUTURE TRENDS

Despite the numerous efforts in recent years, pattern summarization is not yet totally solved. Firstly, all the summarization algorithms were mainly designed to compress only the itemset patterns. But, unfortunately the problem of large output size of FPM algorithms is more severe for patterns, like trees, graphs, etc. Out of the existing algorithms, very few can be generalized to these kinds of patterns, although no attempt has been reported yet. Very recently, Hasan *et al.* proposed an algorithm, called ORIGAMI (Hasan, Chaoji, Salem & Zaki, 2007), which summarize graph patterns by following the idea of representative patterns. Since, finding representative patterns is NP-Hard, they adopted a local optimal algorithm to obtain the representatives. Their algorithm is also a post-processing one; but, they avoided the generation of entire frequent pattern set by finding a smaller collection of maximal patterns through a random walk along the frequent graph lattice. The walk terminates when a user chosen convergence condition is satisfied. This work is very different from all the previous works and it directs the future trends of pattern summarization. Since, generating entire pattern-set is not feasible, an ideal pattern summarization algorithm should avoid the post processing all-together and should find the representative patterns in an online fashion as the patterns are being generated. Some streaming data model can be adopted to solve this kind of problem. This data model also has the added benefit that all pairwise distances between the frequent patterns need not be computed, which, sometimes, is very costly.

CONCLUSION

Pattern summarization is a fascinating research direction with numerous attentions in recent years. To circumvent the problem of enormous output size of FPM algorithms that impede its application, summarization offers an elegant solution. With the invention of new summarization algorithms, many new concepts regarding pattern summarization has also emerged, like similarity, redundancy, significance and etc. These concepts enable to formalize the summarization techniques and many of these, also, provide criteria to evaluate the quality of summarization. This paper provides a gentle overview of different summarization algorithms and these concepts.

REFERENCES

Afrati F., Gionis. A., & Mannila H. (2004). Approximating a Collection of Frequent Sets, *Proceedings of the 10th ACM SIGKDD international conference of Knowledge discovery and Data mining*, 12-19

Agrawal R. & Srikant R. (1994). Fast Algorithms for Mining Association Rules in Large Databases, *Proceedings of the 20th International Conference on Very Large Data Bases*, 487-499

Bayardo, R (1998). Efficiently mining long patterns from databases, *In the Proceedings of ACM SIGMOD Conference.*

Calders T., & Goethals, B. (2007).Non-Derivable Itemset Mining, *Data mining and Knowledge Discovery*, 14(1), 171-206.

Han, J. & Kamber, M. (2001). Data Mining: Concepts and Techniques, 2nd Edition, *Morgan Kaufmann Publications.*

Han, J., Wang, J., Lu, Y. & Tzvetkov, P. (2002). Mining Top-K Frequent Closed Patterns without Minimum Support, *Proceedings of IEE International Conference of Data Mining*, 211-218.

Hasan, M., Chaoji, V., Salem, S., & Zaki, M. (2005). DMTL: A Generic Data Mining Template Library, *in Workshop on Library-Centric Software Design (LCSD'05), with OOPSLA'05 conference, 2005*.

Hasan, M., Chaoji, V., Salem, S., & Zaki, M. (2007). ORIGAMI: Mining Representative Orthogonal Graph Patterns, *Proceedings of IEE International Conference of Data Mining*, 153-162.

Pei J., Dong, G, Zou W, & Han J. (2002). On Computing Condensed Frequent Pattern Bases, *Proceedings of IEE International Conference of Data Mining*, 378-385.

Wang C., & Parthasarathy S. (2006). Summarizing Itemset Patterns Using Probabilistic Models, *Proceedings of the 12th ACM SIGKDD international conference of Knowledge discovery and Data mining*, 730-735.

Xin D., Han J., Yan X., & Cheng H. (2005) Mining Compressed Frequent-Pattern Sets. *Proceedings of the 31st international conference on Very large data Bases*, 709-720.

Xin D., Cheng H., Yan X., & Han J. (2006). Extracting Redundancy-Aware Top-K Patterns, *Proceedings of the 12th ACM SIGKDD international conference of Knowledge discovery and Data mining*, 444-453.

Yan, X. & Han. J. (2002). gSpan: Graph-Based Substructure Pattern Mining, *Proceedings of IEEE International Conference of Data Mining*, 721-724.

Yan X., Cheng H., Han J., & Xin D. (2005). Summarizing Itemset Patterns: A Profile-Based Approach, *Proceedings of the 11th ACM SIGKDD international conference of Knowledge discovery and Data mining*, 314-323.

Zaki, M (2000). Generating Non-Redundant Association Rules, *Proceedings of 6th International Conference of Knowledge discovery and Data mining*, 34-43.

Zaki, M. (2005). Efficiently Mining Frequent Trees in a Forest: Algorithms and Applications, *IEEE Transaction on Knowledge and Data Engineering*, 17(8), 1021-1035.

Zaki, M. (2001). SPADE: An Efficient Algorithm for Mining Frequent Sequences, *Proceedings of Machine Learning Journal, Special Issue on Unsupervised Learning* 42(1/2), 31-60.

KEY TERMS

Closed Pattern: A pattern α is called closed if no pattern β exists, such that $\alpha \subset \beta$ *and transaction_set*(α) = *transaction_set*(β). It is a lossless pattern summarization technique.

Maximal Pattern: A pattern α is called maximal pattern, if no pattern β exists, such that $\alpha \subset \beta$ and β is frequent. It is a lossy summarization technique.

Non-Derivable Pattern: A pattern α is called derivable it its support can be exactly inferred from the support of its sub-patterns based on the inclusion-exclusion principle. Since, inclusion-exclusion principle works only for set, the non-derivable definitions is also applicable only for itemset pattern.

Pattern Profile: Ideally, pattern profile is a compressed representation of a set of very similar patterns, from where the original patterns and their support can be recovered. However, profiles are built using some probabilistic approaches, so the compression is always lossy; therefore, the recovered pattern and their support are usually approximations.

Pattern Significance: Pattern significance measures the importance of a specific pattern in respect to the application domain. For example, in case of itemset pattern which are, usually use to obtain association rule, the pattern significance can be measured in terms of the corresponding rule interestingness. For classification or clustering task, significance of a pattern is the effectiveness of that pattern as a feature to discriminate among different cluster or classes.

Pattern Similarity: Pattern similarity measures the closeness between two patterns. If α and β are two patterns, similarity can be computed as below:

$$sim(\alpha, \beta) = \frac{|\alpha \cap \beta|}{|\alpha \cup \beta|}.$$

Table 1. Comparison of different summarization algorithms, features that have an (↑) symbol are desirable and those that have (↓) are not desirable

Summarization Algorithms	Feature lists							
	Compress ratio↑	Post processing↓	Support preserving↑	Support aware↑	Pattern Redundancy↓	Fault tolerant↓	Coverage considered↑	Generalized↑
Maximal	Moderate	No	No	Yes	High	No	Yes	Yes
Closed	Moderate	No	Yes	Yes	Very High	No	Yes	Yes
Top-k with min length	User defined	No	No	Yes	Low	No	No	Yes
Fault-tolerant Top-k	User defined	Yes	No	No	Very Low	Yes	Yes	No
Redundancy-aware top-k	User defined	Yes	No	No	Very Low	No	Yes	Yes
Non-derivable	Moderate	No	Yes	Yes	High	No	Yes	No
Cluster based representatives	User defined	Yes	No	Yes	Low	No	Yes	Yes
Pattern profile	High	Yes	No	Yes	Very Low	Yes	No	No
Probabilistic profile	High	Yes	No	No	Very Low	Yes	No	No

The set union and intersection in the above definition can be extended naturally for complex pattern like, tree or graph. For instance, for graph pattern, intersection between two patterns can be the maximal common sub-graph between them. Generally, pattern similarity ranges from [0, 1]. Pattern distance is computed as 1–*Pattern Similarity.* Distance obeys the metric properties, like, $distance(\alpha,\alpha) = 0$ *and* $distance(\alpha,\beta) + distance(\beta,\gamma) \geq distance(\alpha,\gamma)$.

Pattern Summarization: FPM algorithms produce large results which is difficult for the end user to interpret. To circumvent the problem, frequent patterns are compressed to a smaller set, where the patterns are non-redundant, discriminative and representative. Effective summarizations are, generally, lossy *i.e.* the support information of the compressed patterns can not be recovered exactly. Different flavors of summarization techniques exist. Pattern profile is one of the summarization techniques.

Supporting Imprecision in Database Systems

Ullas Nambiar
IBM India Research Lab, India

INTRODUCTION

A query against incomplete or imprecise data in a database[1], or a query whose search conditions are imprecise can both result in answers that do not satisfy the query completely. Such queries can be broadly termed as ***imprecise queries***. Today's database systems are designed largely for precise queries against a database of precise and complete data. Range queries (*e.g., Age BETWEEN 20 AND 30*) and disjunctive queries (*e.g., Name="G. W. Bush" OR Name="George Bush"*) do allow for some imprecision in queries. However, these extensions to precise queries are unable to completely capture the expressiveness of an imprecise query. Supporting imprecise queries (*e.g., Model like "Camry" and Price around "$15000")* over databases necessitates a system that integrates a similarity search paradigm over structured and semi-structured data. Today's relational database systems, as they are designed to support precise queries against precise data, use such precise access support mechanisms as indexing, hashing, and sorting. Such mechanisms are used for fast selective searches of records within a table and for joining two tables based on precise matching of values in join fields in the tables. The imprecise nature of the search conditions in queries will make such access mechanisms largely useless. Thus, supporting imprecise queries over existing databases would require adding support for imprecision within the query engine and meta-data management schemes like indexes.

Extending a database to support imprecise queries would involve changing the query processing and data storage models being used by the database. But, the fact that databases are generally used by other applications and therefore must retain their behaviour could become a key inhibitor to any technique that relies on modifying the database to enable support for imprecision. For example, changing an airline reservation database will necessitate changes to other connected systems including travel agency databases, partner airline databases etc. Even if the database is modifiable, we would still require a domain expert and/or end user to provide the necessary distance metrics and domain ontology. Domain ontologies do not exist for all possible domains and the ones that are available are far from being complete. Therefore, a feasible solution for answering imprecise queries should neither assume the ability to modify the properties of the database nor require users (both lay and expert) to provide much domain specific information.

BACKGROUND

The problem of supporting imprecise queries has already attracted considerable interest from researchers including those in fuzzy information systems (Morrisey, 1990), cooperative query answering and query generalization (Motro, 1998). More recent efforts have focused on supporting imprecise queries over relational databases by introducing ADTs (abstract data types) – for allowing storage and retrieval of non-standard data such as images, documents etc., and extending the query processor with functions for measuring similarity between tuples (Aditya et al, 2002; Ortega-Binderberger, 2003). Recently, work has been done on providing ranked answers to queries over a relational database (Bruno, Gravano and Marian, 2002). However, all the proposed approaches for answering imprecise queries require large amounts of domain specific information either pre-estimated or given by the user of the query. Unfortunately, such information is hard to elicit from the users. Further, some approaches require changing the data models and operators of the underlying database. A recent survey outlining challenges in integrating DB (database) and IR (information retrieval) technologies discusses the pros and cons of four possible alternatives for combining the two models (Chaudhuri, Ramakrishnan and Weikum, 2005).

MAIN FOCUS

The task of supporting imprecise queries over a database can borrow several ideas from efforts at integrating DB & IR systems. However, the key difference is that the focus is only on bringing the IR style search and retrieval model to DB systems with the underlying data continuing to be structured. The motivation should be to reduce the burden on the user by striving to satisfy the users' imprecisely defined need (query) with minimal additional information from user. Given that the database contains tuples generated by humans, they must capture some amount of real-world semantics e.g., relationships between attributes (features of the domain), similarity between values, etc. Solutions for supporting imprecise queries should focus on using the inherent semantics to help the user in extracting relevant information. Any solution then should be judged based on how good is its estimation of the user need based on the query and underlying data, i.e., *How closely does it model the user's notion of relevance by using only the information available in the database?*

A domain independent solution for supporting imprecise queries over autonomous databases is given in Nambiar and Kambhampati, 2006. The solution unites the DB and IR technologies by bringing the similarity searching/ranked retrieval paradigm from IR systems into the structured, type-rich access paradigm of databases. Answers are ranked according to the degree of relevance automatically estimated using domain-independent similarity functions that can closely approximate the subjective interpretation of the user. An intuitive model for measuring the similarity between the query and the answer tuples is by measuring the similarity of common attributes between them. Not all attributes will be equally relevant to the user. Therefore, providing a ranked set of answers with minimal user input and domain related metrics would require techniques to automatically learn the similarity between values binding each attribute and also the importance of every attribute in a relation.

Measuring Value Similarity

A database system supporting imprecise queries must provide information about how close an answer tuple is to the given imprecise query. Two tuples (a selection query can be seen as a tuple with few missing values) are considered similar if they have *syntactical similarity* (e.g., same subset of attributes are bound in both queries, stems of a common word bind an attribute, etc) or if the binding values are *semantically similar*. Semantic similarity is a concept whereby a set of words (attribute values) are assigned a metric based on the closeness of their meaning. Similar words can be considered semantically related by virtue of their *synonymy* (e.g., bank – trust company), but dissimilar entities may also be semantically related by lexical relationships such as *meronymy* (e.g., car - wheel) and *antonymy* (e.g., hot - cold), or just by any kind of functional relationship or frequent association (e.g., pencil - paper, penguin - Antarctica, rain - flood). The definition of synonym by Leibniz - *"synonyms are words that are interchangeable in some contexts"* is considered more realistic. This definition forms the basis of the similarity estimation model developed in Nambiar and Kambhampati, 2006. Given a database of tuples, the authors assume that binding values that are semantically similar have similar distributional behaviour. Under this assumption, they treat the values that co-occur near a value as constituting features that describe the context in which the given value appears in the database. The semantic similarity between two values is then computed in terms of how similar is their contexts. This is accomplished by building a structure consisting of bags of words for all attributes in the relation not bound by the two values being considered. Only values that co-occur with the value under consideration is added to the bag of words. The similarity between two values is then computed as the level of commonality between the bags of words describing them (see Nambiar and Kambhampati, 2006 for details).

Learning Attribute Importance

Often users would like to see only the top-k answers to a query. To provide ranked answers to a query, we must combine similarities shown over distinct attributes of the relation into an overall similarity score for each tuple. Specifically, a measure of importance for the similarity shown over any attribute in the context of a given query may be necessary to determine the best k matches. While this measure may vary from user to user, most users usually are unable to correctly quantify the importance they ascribe to an attribute. In theory the tuples most similar to query will have differences only in the *least important attribute*. Nambiar and Kambhampati, 2006, define the least important at-

tribute as the attribute whose binding value, when changed, has minimal effect on values binding other attributes. They provide an algorithm for automatically measuring the importance of an attribute based on Approximate Functional Dependencies (Huhtala et al, 1998). Causal Bayesian Networks (Cooper, 1997) and Causal Association Rules (Silverstein et al, 1998) are alternative techniques that are useful in learning causal relationships among attributes.

FUTURE DIRECTIONS

This chapter has focussed only on answering imprecise queries over a single autonomous database and provides a solution (Nambiar and Kambhampati, 2006) for the same. But more than one database may project a given relation. Even if all these systems were extended to support imprecise queries, it is not feasible for the user to query all of them to obtain most of the relevant answers. Supporting imprecise queries over multiple sources will involve determining the number of relevant tuples each database is likely to contain for every possible imprecise query. Since learning and storing the statistics for all possible imprecise queries is infeasible, one will have to learn the same for classes of imprecise queries. The second step is to provide an efficient plan for accessing the databases, based on their likelihood of providing the best remaining set of answers. Nie et al, 2002, describe a solution for efficiently obtaining tuples from an overlapping set of autonomous databases. Some of those ideas could be used to tackle the problems in answering imprecise queries over a set of overlapping databases. Yu et al, 2003, give two algorithms for merging results from ranked databases. However, no overlap among databases is considered.

CONCLUSION

An increasing amount of database researchers envision the need for supporting imprecise queries over database systems. Such support would greatly enhance the ability of non-expert users to extract relevant information from databases. For example, users working with scientific databases such as the Web accessible repositories in archeology and biology can greatly benefit by being able to quickly explore the knowledge contained without having to worry about forming the right query to express their need. Supporting imprecise queries over autonomous databases entails overcoming critical challenges like developing techniques for efficiently extracting relevant tuples and measuring the similarity of the answers to the query. Overcoming these challenges necessitates developing techniques for estimating the importance to be ascribed to each attribute and for measuring the semantic similarity between categorical values.

REFERENCES

Morrisey, J.M. (1990). Imprecise Information and Uncertainty in Information Systems. *ACM Transactions on Information Systems*, 8:159–180.

Motro, A. (1998). Vague: A user interface to relational databases that permits vague queries. *ACM Transactions on Office Information Systems*, 6(3):187–214.

Aditya, B., Bhalotia, G., Chakrabarti, S., Hulgeri, A., Nakhe, C., Parag and S. Sudarshan (2002). BANKS: Browsing and Keyword Searching in Relational Databases. *In VLDB.*

Bruno, N., Gravano, L., and Marian, A. (2002). Evaluating Top-K Queries over Web-Accessible Databases. *In proceedings of ICDE.*

Ortega-Binderberger, M. (2003). Integrating Similarity Based Retrieval and Query Refinement in Databases. *PhD thesis, UIUC.*

Haveliwala, T., Gionis, A., Klein, D., and Indyk, P. (2002). Evaluating Strategies for Similarity Search on the Web. *In proceedings of WWW.*

Nambiar, U. and Kambhampati, S. (2006). Answering Imprecise Queries over Autonomous Web Databases. *In proceedings of ICDE.*

Chaudhuri, S., Ramakrishanan, R. and Weikum, G. (2005). Integrating DB and IR Technologies: What is the Sound of One Hand Clapping? *In proceedings of CIDR.*

Huhtala, Y., Krkkinen, J., Porkka, P., and H. Toivonen (1998). Efficient Discovery of Functional and Approximate Dependencies Using Partitions. *In proceedings of ICDE.*

Cooper, G., (1997). A simple constraint-based algorithm for efficiently mining observational databases for causal relationships. *Data Mining and Knowledge Discovery, 2*.

Yu, C., Philip, G., and Meng, W. (2003). Distributed Top-N Query Processing with Possibly Uncooperative Local Systems. *In proceedings of VLDB.*

Silverstein, C., Brin, S., Motwani, R., and Ullman, J. (1998). Scalable Techniques for Mining Causal Structures. In proceedings of VLDB.

Nie, Z., Nambiar, U., Vaddi, S., and Kambhampati, S., (2002). Mining Coverage Statistics for Websource Selection in a Mediator. *In proceedings of CIKM.*

KEY TERMS

Attribute: An attribute is a template for possible values and set of functions and operators that operate on these values and define the behavior.

Database: An information set with a regular structure. Databases based on the *relational model* are known as *relational databases*.

Data Integration System: A system that provides an automated method for querying across multiple heterogeneous databases in a uniform way. In a data integration system, the user asks a query over a common schema, called *mediated schema or global schema*, and the system reformulates this into a query over the *local schema* of the data sources.

Imprecise Query: A user query that does not insist on exact match (and only requires data closely matching the query constraint) is an imprecise query.

Information Retrieval (IR): The art and science of searching for information from free-form natural language text. An alternate definition is that of a process for identifying unstructured records satisfying a user query.

Precise Query: A user query that requires data exactly matching the query constraint is a precise query. A precise query contains only crisp conditions over the attributes.

Tuple: A tuple is a set of attributes, which are ordered pairs of domain and value. A *relation* is an unordered set of tuples.

ENDNOTE

[1] The term database is used to refer to all types of structured data stores including data warehouses.

A Survey of Feature Selection Techniques

Barak Chizi
Tel-Aviv University, Israel

Lior Rokach
Ben-Gurion University, Israel

Oded Maimon
Tel-Aviv University, Israel

INTRODUCTION

Dimensionality (i.e., the number of data set attributes or groups of attributes) constitutes a serious obstacle to the efficiency of most data mining algorithms (Maimon and Last, 2000). The main reason for this is that data mining algorithms are computationally intensive. This obstacle is sometimes known as the "curse of dimensionality" (Bellman, 1961).

The objective of Feature Selection is to identify features in the data-set as important, and discard any other feature as irrelevant and redundant information. Since Feature Selection reduces the dimensionality of the data, data mining algorithms can be operated faster and more effectively by using Feature Selection. In some cases, as a result of feature selection, the performance of the data mining method can be improved. The reason for that is mainly a more compact, easily interpreted representation of the target concept.

The filter approach (Kohavi , 1995; Kohavi and John ,1996) operates independently of the data mining method employed subsequently -- undesirable features are filtered out of the data before learning begins. These algorithms use heuristics based on general characteristics of the data to evaluate the merit of feature subsets. A sub-category of filter methods that will be refer to as rankers, are methods that employ some criterion to score each feature and provide a ranking. From this ordering, several feature subsets can be chosen by manually setting

There are three main approaches for feature selection: wrapper, filter and embedded.

The wrapper approach (Kohavi, 1995; Kohavi and John,1996), uses an inducer as a black box along with a statistical re-sampling technique such as cross-validation to select the best feature subset according to some predictive measure.

The embedded approach (see for instance Guyon and Elisseeff, 2003) is similar to the wrapper approach in the sense that the features are specifically selected for a certain inducer, but it selects the features in the process of learning.

BACKGROUND

Feature selection algorithms search through the space of feature subsets in order to find the best subset. This subset search has four major properties (Langley, 1994): starting point, search organization, evaluation strategy, and stopping criterion.

Starting Point: Selecting a point in the feature subset space from which to begin the search can affect the direction of the search.

Search Organization: A comprehensive search of the feature subspace is prohibitive for all but a small initial number of features.

Evaluation Strategy: How feature subsets are evaluated (filter, wrapper and ensemble).

Stopping Criterion: A feature selector must decide when to stop searching through the space of feature subsets.

FEATURE SELECTION TECHNIQUES

This section provides a survey of techniques for each strategy described on previous sections.

Feature Filters

The filter methods were the earliest approaches for feature selection. All filter methods use general properties of the data in order to evaluate the merit of feature subsets. As a result, filter methods are generally much faster and practical than wrapper methods, especially for using it on data of high dimensionality. Detailed experiments for each method presented below can be found in Hall's work (1999) and on Liu and Motoda (1998) book on Feature Selection.

FOCUS

Almuallim and Dietterich (1991) describe an algorithm originally designed for Boolean domains called FOCUS. FOCUS exhaustively searches the space of feature subsets until every combination of feature values is associated with one value of the class. After selecting the subset, it passed to ID3 (Quinlan, 1986), which constructs a decision tree.

LVF

Similar algorithm to FOCUS is LVF(Liu and Setiono, 1996) describe. LVF is consistency driven and can handle noisy domains if the approximate noise level is known a- priori. Every round of execution LVF generates a random subset from the feature subset space. If the chosen subset is smaller than the current best subset, the inconsistency rate of the dimensionally reduced data described by the subset is compared with the inconsistency rate of the best subset. If the subset is at least as consistent as the best subset, the subset replaces the best subset.

An Information Theoretic Feature Filter

Koller and Sahami (1996) described a feature selection algorithm based on information theory and probabilistic reasoning. The rationale behind this technique is that, since the goal of an induction algorithm is to estimate the probability distributions over the class values, given the original feature set, feature subset selection should attempt to remain as close to these original distributions as possible.

An Instance Based Approach to Feature Selection – RELIEF

RELIEF (Kira and Rendell, 1992) uses instance based learning to assign a relevance weight to each feature. The weight for each feature reflects its ability to single out the class values. The Features are ranked by its weights and chosen by using a user- specified threshold. RELIEF randomly choosing instances from the training data. For every instance RELIEF samples the nearest instance of the same class (nearest hit) and finds the opposite class (nearest miss). The weight for each feature is updating according to how well its values differentiate the sampled instance from its nearest hit and nearest miss. Feature will gain a high weight if it differentiates between instances from different classes and has the same value for instances of the same class.

Simba and G-Flip

Gilad-Bachrach et al. (2004), introduced a new approach called SIMBA (Iterative Search Margin Based Algorithm), which outperforms RELIF. This approach introduces the idea of measuring the quality of a set of features by the margin it induces. To overcome the drawback of iterative search, Gilad-Bachrach et al (2004), present A Greedy Feature Flip Algorithm called G-Flip. The G-Flip is a greedy search algorithm for maximizing the margin function

of a subset. The algorithm constantly iterates over the feature set and updates the set of chosen features. Each iteration G-Flip decides to eliminate or include the current feature to the selected subset by evaluating the margin with and without this feature.

Using Traditional Statistics Techniques

Mallows Cp (Mallows. 1973)

This method minimizes the mean square error of prediction:

$$C_p = \frac{RSS_\gamma}{\hat{\sigma}^2_{FULL}} + 2q_\gamma - n \qquad (1)$$

Where, RSS_γ is the residual sum of squares for the γ'*th* model and $\hat{\sigma}^2_{FULL}$ is the usual unbiased estimate of the variance based on the full model. The goal is to find the subset which has minimum *Cp*.

AIC, BIC and F Ratio

AIC (for Akaike Information Criterion) and BIC (for Bayesian Information Criterion) are criteria for choosing a subset of features. Letting $\hat{l}_g$ denote the maximum log likelihood of the γ'*th* model, AIC selects the model which maximizes ($\hat{l}_g - q_\gamma$) whereas BIC selects the model which maximizes ($\hat{l}_g$ - (logn)$q_\gamma/2$).

For the linear model, many of the popular selection criteria are a penalized sum of squares criterion that can provide a unified framework for comparisons. This criterion selects the subset model that minimizes:

$$RSS_\gamma / \hat{\sigma}^2 + Fq_\gamma \qquad (2)$$

Where *F* is a preset "dimensionality penalty". The above penalizes $RSS_\gamma / \hat{\sigma}^2$ by *F* times q_γ, the dimension of the γth model. AIC and minimum *Cp* are equivalent, corresponding to *F*=2, and BIC is obtained by *F* =*logn*. Using a smaller penalty, AIC and minimum *Cp* will select larger models than BIC (unless n is very small), (George, 2000).

Principal Component Analysis (PCA)

Principal component analysis (PCA) (known also as the singular value decomposition (SVD), the Karhunen-Loeve transform, the Hotelling transform, and the empirical orthogonal function (EOF) method.), is linear dimension reduction technique (Jackson, 1991). PCA based on the covariance matrix of the variables and it is a second-order method. PCA seeks to reduce the dimension of the data by finding a few orthogonal linear combinations (the PCs) of the original features with the largest variance. The first PC, s_1, is the linear combination with the largest variance. We have $s_1 = x^T w_1$, where the *p*-dimensional coefficient vector $w_1 = (w_{1,1},\ldots,w_{1,p})^T$ solves:

$$w_1 = \arg\max_{\|w=1\|} Var\left\{x^T w\right\} \qquad (3)$$

The second PC is the linear combination with the second largest variance and orthogonal to the first PC, and so on. There are as many PCs as the number of the original features. For many datasets, the first several PCs explain most of the variance, so that the rest can be ignored with minimal loss of information.

Factor Analysis (FA)

Factor analysis (FA) is a linear method, based on the second-order data summaries. FA assumes that the measured features depend on some unknown factors. Typical examples include features defined as various test scores of individuals; as such scores are thought to be related to a common "intelligence" factor. The goal of FA is to find out such relations, and thus can be used to reduce the dimension of datasets following the factor model.

Projection Pursuit

Projection pursuit (PP) is a linear method which is more computationally intensive than second-order methods. Given a projection index that defines the merit of a direction, the algorithm looks for the directions that optimize that index. As the Gaussian distribution is the least interesting distribution, projection indices usually measure some aspect of non-Gaussianity.

Feature Wrappers

The Wrapper strategy for feature selection uses an induction algorithm to evaluate feature subsets. The motivation for this strategy is that the induction method that will eventually use the feature subset should provide a better predictor of accuracy than a separate measure that has an entirely different inductive bias (Langley, 1994).

Feature wrappers are often better than filters. The reason for that is that they are tuned to the specific interaction between an induction algorithm and its training data. Nevertheless, they tend to be much slower than feature filters because they must repeatedly perform the induction algorithm. Detailed experiments for each method presented below can be found in Hall's work (1999) and on Liu and Motoda (1998) book on Feature Selection.

Wrappers for Decision Tree Learners

John, Kohavi, and Pfleger (1994) presented the wrapper as a general framework for feature selection in machine learning. This framework has two degrees of feature relevance definitions that used by the wrapper to discover relevant features. A feature X_i is said to be strongly relevant to the target concept(s) if the probability distribution of the class values, given the full feature set, changes when X_i is eliminated. A feature X_i is said to be weakly relevant if it is not strongly relevant and the probability distribution of the class values, given some subset which contains X_i, changes when X_i is removed. All features that are not strongly or weakly relevant are irrelevant.

Vafaie & De Jong (1995) and Cherkauer & Shavlik (1996) have both applied genetic search strategies in a wrapper framework in order to improve the performance of decision tree learners.

Wrappers for Instance Based Learning

OBLIVION (Langley and Sage, 1994) is a wrapper for instance based learning. OBLIVION performs backward elimination of features by using an oblivious decision tree as the induction algorithm. Moore and Lee (1994) suggest a similar approach to augmenting nearest neighbor algorithm but their system uses leave one out instead of *k*-fold cross validation and focuses on improving the prediction of numeric rather than discrete classes.

Domingos (1997) describes a context sensitive wrapper approach to feature selection for instance based learners. The motivation for the approach is that there may be features that are either relevant in only a restricted area of the instance space and irrelevant elsewhere, or relevant given only certain values (weakly interacting) of other features and otherwise irrelevant. In either case, when features are estimated globally (over the instance space). The irrelevant aspects of these sorts of features may overwhelm their entire useful aspects for instance based learners.

Wrappers for Bayes Classifiers

Langley and Sage (1994) note that the classifier performance on domains with redundant features can be improved by removing such features. In order to select features for use with naïve Bayes A forward search strategy is employed. The rationale for using a forward search is that it should immediately detect dependencies when redundant attributes are added.

Pazzani (1995) combines feature selection and simple constructive induction in a wrapper framework for improving the performance of naive Bayes. The algorithm considers not only to add features to the

current subset, but also to creating new features by joining one of the as yet unselected features with each of the selected features in the subset. More than that, the algorithm considers both deleting individual features and replacing pairs of features with a joined feature.

Provan and Singh (1996) have applied the wrapper to select features from which to construct Bayesian networks. Their results showed that while feature selection did not improve accuracy over networks constructed from the full set of features, the networks created after feature selection were considerably smaller and faster to learn.

Feature Ensemble

The main idea of ensemble methodology is to combine a set of models, each of which solves the same original task, in order to obtain a better composite global model, with more accurate and reliable estimates or decisions than can be obtained from using a single model. Some of the drawbacks of wrappers and filters can be solved by using ensemble (Rokach *et al.*, 2007). As mentioned above filters perform less than wrappers. Due to the voting process, noisy results are filtered. Secondly, the drawback of wrappers which "cost" computing time is solved by operating bunch of filters.

FUTURE TRENDS

All feature selection techniques mentioned above assume that some of features of the initial subset are redundant or irrelevant. But in some cases data mining techniques creates models that do not have any redundant or irrelevant features. Theoretical and empirical observations for this situation can be found on Chizi and Maimon (2002) work on dimensionality reduction of high dimensional data sets. It seems that when using any feature selector there is a point, (which called by the authors "Phase Transition Point") that further removing of features will cause significant damage to the accuracy of the model.

CONCLUSION

Feature selection is a common issue on statistics, pattern recognition and machine learning. This Chapter summarizes the main techniques and approaches for feature selection. The aim of any feature selector is to identify features in the data-set as important, and discard any other feature as irrelevant and redundant information. All three strategies mentioned on this chapter (Filter, Wrapper and Ensemble) contain a wide variety of techniques. The usefulness of each methods mainly depends on it ability to provide swift and accurate model. The following table summarizes the main properties of each strategy by ranking from 1 (best in the category) to 3 (worst in the category) (see Exhibit 1).

Exhibit 1.

Category	Filter	Wrapper	Ensemble
Computational Cost	1	3	2
Robustness	3	1	2
Best Accuracy	3	2	1
Simplicity	1	3	2

REFERENCES

Almuallim, H. and Dietterich, T. G. (1991). Learning with many irrelevant features. In *Proceedings of the Ninth National Conference on Artificial Intelligence,* 547– 542. MIT Press.

Bellman, R. (1961) Adaptive Control Processes: A Guided Tour, Princeton University Press.

Cherkauer, K. J. and Shavlik, J. W. (1996) Growing simpler decision trees to facilitate knowledge discovery. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. AAAI Press.

Chizi, B. and Maimon, O. (2002). On Dimensionality Reduction of High Dimensional Data Sets, In *Frontiers in Artificial Intelligence and Applications*. pp. 230-236. IOS press

Domingos, P. (1997). Context- sensitive feature selection for lazy learners. *Artificial Intelligence Review*, (11): 227– 253.

Gilad-Bachrach, R., Navot, A. and Tishby. (2004) N. Margin based feature selection - theory and algorithms. In *Proceeding of the 21'st International Conferenc on Machine Learning.*

Guyon I. and Elisseeff A. (2003), An introduction to variable and feature selection, *Journal of Machine Learning Research* (3), pp. 1157-1182.

Hall, M. (1999), *Correlation- based Feature Selection for Machine Learning*. PhD thesis University of Waikato.

Hu, X. (2001) Using Rough Sets Theory and Database Operations to Construct a Good Ensemble of Classifiers for Data Mining Applications. *Proceeding of the ICDM01*. 233-240.

Jackson, J. (1991) A User's Guide to Principal Components. New York: John Wiley and Sons.

John, G. H. Kohavi, R. and Pfleger, P. (1994). Irrelevant features and the subset selection problem. *In Machine Learning: Proceedings of the Eleventh International Conference*. Morgan Kaufmann.

Kira, K. and Rendell, L. A. (1992). A practical approach to feature selection. In *Machine Learning: Proceedings of the Ninth International Conference*.

Kohavi R. and John, G. (1996). Wrappers for feature subset selection. *Artificial Intelligence*, special issue on relevance, 97(1– 2): 273– 324.

Kohavi, R. (1995). *Wrappers for Performance Enhancement and Oblivious Decision Graphs*. PhD thesis, Stanford University.

Kohavi, R. and Sommerfield, D. (1995). Feature subset selection using the wrapper method: Overfitting and dynamic search space topology. In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*. AAAI Press.

Koller, D. and Sahami, M. (1996). Towards optimal feature selection. In *Machine Learning: Proceedings of the Thirteenth International Conference on machine Learning*. Morgan Kaufmann.

Langley, P. and Sage, S. (1994). Scaling to domains with irrelevant features. In R. Greiner, (Ed.), *Computational Learning Theory and Natural Learning Systems,* volume 4. MIT Press.

Langley, P. (1994). Selection of relevant features in machine learning. In *Proceedings of the AAAI Fall Symposium on Relevance*. AAAI Press.

Liu, H. and Motoda, H. (1998) *Feature Selection for Knowledge Discovery and Data-Mining*, Kluwer Academic Publishers.

Liu, H. and Setiono, R. (1996) A probabilistic approach to feature selection: A filter solution. In Machine Learning: *Proceedings of the Thirteenth International Conference on Machine* Learning. Morgan Kaufmann.

Maimon, O. and Last, M. (2000). *Knowledge Discovery and Data Mining – The Info-Fuzzy Network (IFN) Methodology*, Kluwer.

Mallows, C. L. (1973) Some comments on Cp . *Technometrics* 15, 661- 676.

Moore, A. W. and Lee, M. S. (1994) Efficient algorithms for minimizing cross validation error. In *Machine Learning: Proceedings of the Eleventh International Conference*. Morgan Kaufmann.

Moore, A. W. Hill, D. J. and Johnson, M. P. (1992) An empirical investigation of brute force to choose features, smoothers and function approximations. In S. Hanson, S. Judd, and T. Petsche, (Eds.), *Computational Learning Theory and Natural Learning Systems*, (3). MIT Press.

Pazzani, M. (1995) Searching for dependencies in Bayesian classifiers. In *Proceedings of the Fifth International Workshop on AI and Statistics.*

Provan, G. M. and Singh, M. (1996). Learning Bayesian networks using feature selection. In D. Fisher and H. Lenz, (Eds.), *Learning from Data, Lecture Notes in Statistics*, pp. 291– 300. Springer-Verlag, New York.

Quinlan, J.R. (1986). Induction of decision trees. *Machine Learning*, 1: 81– 106.

Quinlan, J.R. (1993). *C4.5 Programs for machine learning*. Morgan Kaufmann, Los Altos, California.

Rissanen, J. (1978) Modeling by shortest data description. *Automatica*, 14: 465–471.

Rokach L., Chizi B., Maimon O. (2007) A Methodology for Improving the Performance of Non-ranker Feature Selection Filters, International Journal of Pattern Recognition and Artificial Intelligence, Vol. 21, No. 5: 809–830

Setiono, R. and Liu, H. (1995) Chi2: Feature selection and discretization of numeric attributes. In *Proceedings of the Seventh IEEE International Conference on Tools with Artificial Intelligence*.

Singh, M. and Provan, G. M. (1996). Efficient learning of selective Bayesian classifiers. In Machine Learning: *Proceedings of the Thirteenth International network Conference on Machine Learning*. Morgan Kaufmann.

Skalak, B. (1994). Prototype and feature selection by sampling and random mutation hill climbing algorithms. In *Machine Learning: Proceedings of the Eleventh International Conference*. Morgan Kaufmann.

Tumer, K. and Ghosh J. (1999). Error Correlation and Error Reduction in Ensemble Classifiers, *Connection Science*, Special issue on combining artificial neural networks: ensemble approaches, 8 (3-4): 385-404.

Vafaie, H. and De Jong, K. (1995). Genetic algorithms as a tool for restructuring feature space representations. In *Proceedings of the International Conference on Tools with A. I.* IEEE Computer Society Press.

KEY TERMS

Attribute: A quantity describing an instance. An attribute has a domain defined by the attribute type, which denotes the values that can be taken by an attribute.

Classifier: A structured model that maps unlabeled instances to finite set of classes

Feature Selection: A process aims to identify the important attributes in a given database and discard any other feature as irrelevant and redundant information.

Filter Techniques : These techniques use heuristics based on general characteristics of the data to evaluate the merit of feature subsets.

Induction Algorithm: An algorithm that takes as input a certain set of instances and produces a model that generalizes these instances.

Instance: A single object of the world from which a model will be learned, or on which a model will be used.

Ranker Techniques: Rankers are methods that employ some criterion to score each feature. From

this ordering, several feature subsets can be chosen, either manually or by setting a threshold.

Wrapper Techniques: These techniques use induction algorithms along with statistical re-sampling techniques such as cross-validation to select the best feature subset for this specific learning algorithm.

Survival Data Mining

Qiyang Chen
Montclair State University, USA

Ruben Xing
Montclair State University, USA

Richard Peterson
Montclair State University, USA

Dajin Wang
Montclair State University, USA

INTRODUCTION

Survival analysis (SA) consists of a variety of methods for analyzing the timing of events and/or the times of transition among several states or conditions. The event of interest can only happen at most once to any individual or subject. Alternate terms to identify this process include *Failure Analysis* (FA), *Reliability Analysis* (RA), *Lifetime Data Analysis* (LDA), *Time to Event Analysis* (TEA), *Event History Analysis* (EHA), and *Time Failure Analysis* (TFA) depending on the type of application the method is used for (Elashoff, 1997). Survival Data Mining (SDM) is a new term being coined recently (SAS, 2004). There are many models and variations on the different models for SA or failure analysis. This chapter discusses some of the more common methods of SA with real life applications. The calculations for the various models of SA are very complex. Currently, there are multiple software packages to assist in performing the necessary analyses much more quickly.

BACKGROUND

The history of SA can be roughly divided into four periods. The four periods are the Grauntian, Mantelian, Coxian and Aalenian paradigm (Harrington, 2003). The first paradigm dates back to the 17th century with Graunt's pioneering work which attempted to understand the distribution for the length of human life (Holford, 2002) through life tables. During World War II, early life table's analysis led to reliability studies of equipment and weapons and was called TFA.

The *Kaplan-Meier method,* a main contribution during the second paradigm, is perhaps the most popular means of SA. In 1958, a paper by Kaplan and Meier in the Journal of the American Statistical Association "brought the analysis of right-censored data to the attention of mathematical statisticians..." (Oakes, 2000, p.282). The Kaplan-Meir' *product limit method* is a tool used in SA to plot survival data for a given sample of a survival study. Hypothesis testing continued on these missing data problems until about 1972. Following the introduction by Cox of the *proportional hazards model*, the focus of attention shifted to examine the impact of survival variables (covariates) on the probability of survival through the period of third paradigm. This survival probability is known within the field as the "hazard function."

The fourth and last period is the Aalenian paradigm as Statsoft (2003) claims. Aalen used a martingale approach (exponential rate for counting processes) and improved the statistical procedures for many problems arising in randomly censored data from biomedical studies in the late seventies of last century.

MAIN FOCUS

The two biggest pitfalls in SA is the considerable variation in the risk across the time interval which demonstrates the need for shorter time intervals and censoring. Censored observations occur when there is a loss of observation. This most often arises when subjects withdraw or are lost from follow-up before the completion of the study. The effect of censoring often

renders a bias within studies based upon incomplete data or partial information on survival or failure times.

There are four basic approaches for the analysis of censored data: complete data analysis; the imputation approach; analysis with dichotomized data; and the likelihood-based approach (Leung et al., 1997). The most effective approach to censoring problems is to use methods of estimation that adjust for whether or not an individual observation is censored. These "likelihood-based approaches" include the *Kaplan-Meier estimator* and the *Cox-regression*, both popular methodologies. The *Kaplan-Meier estimator* allows for the estimation of survival over time even for populations that include subjects who enter at different times or drop out.

Having discovered the inapplicability of multiple regression techniques due to the distribution (exponential vs. normal) and censoring, Cox assumed "a multiplicative relationship between the underlying hazard function and the log-linear function of the covariates" (Statsoft, 2003) and arrived at the assumption that the underlying hazard rate (rather than survival time) is a function of the independent variables (covariates) by way of a nonparametric model.

As SA emerged and became refined through the periods, it is evident even from the general overview herein that increasingly more complex mathematical formulas were being applied. This was done in large measure to account for some of the initial flaws in the research population (i.e. censoring), to provide for the comparison of separate treatments, or to take entirely new approaches concerning the perceived distributions of the data. As such, the calculations and data collection for the various models of SA became very complex requiring the use of equally sophisticated computer programs.

In that vein, software packages capable of performing the necessary analyses have been developed and include but are not limited to: SAS/STAT software (compares survival distributions for the event-time variables, fits accelerated failure time models…and performs regression analysis based on the proportional hazards model) (SAS, 2003). Also available is software from NCSS 2004 Statistical Analysis System (NCSS, 2003) and SPSS (Pallant, 2007).

Multiple-Area Applications

The typical objective of SA in demography and medical research centers on clinical trials designed to evaluate the effectiveness of experimental treatments; the modeling of disease progression in an effort to take preemptive action; and also for the purpose of estimating disease prevalence within a population. The fields of engineering and biology found applicability of SA later. There is always a need for more data analysis. The information gained from a successful SA can be used to make estimates on treatment effects, employee longevity, or product life. As SA went through more advanced stages of development it started to be also used in business related fields like economics and social sciences. With regards to a business strategy, SA can be used to predict, and thereby improve upon, the life span of manufactured products or customer relations. For example, by identifying the timing of "risky behavior patterns" (Teredata, 2003) that lead to reduced survival probability (ending the business relationship) in the future, a decision can be made to select the appropriate marketing action and its associated cost. Lo, MacKinlay and Zhang (2002) of MIT Sloan School of Management developed and estimated an econometric model of limit-order execution times. They estimated versions for time-to-first-fill and time-to-completion for both buy and sell limit orders, and incorporated the effects of explanatory variables such as the limit price, limit size, bid/offer spread, and market volatility. Through SA of actual limit-order data, they discovered that execution times are very sensitive to the limit price, but are not sensitive to limit size. Hypothetical limit-order executions, constructed either theoretically from first-passage times or empirically from transactions data, are very poor proxies for actual limit-order executions.

Blandón (2001) investigated the timing of foreign direct investment in the banking sector which, among other things, leads to differential benefits for the first entrants in a foreign location, and to problem of reversibility. When uncertainty is considered, the existence of some ownership–location–internalization advantages can make foreign investment less reversible and/or more delayable. Such advantages are examined and a

model of the timing of foreign direct investment specified. The model is then tested for a case using duration analysis.

In many industries, alliances have become the organization model of choice. Having used data from the *Airline Business* annual surveys of airline alliances, Gudmundsson and Rhoades (2001) tested a proposed typology predicting survival and duration in airline alliances. They classified key activities of airline alliances by their level of complexity and resource commitment in order to suggest a series of propositions on alliance stability and duration. The results of their analysis indicate that alliances containing joint purchasing and marketing activities had lower risk of termination than alliances involving equity.

Kimura and Fujii (2003) conducted a Cox-type SA of Japanese corporate firms using census-coverage data. A study of exiting firms confirmed several characteristics of Japanese firms in the 1990s. They found that in order to increase the probability of survival, an efficient concentration on core competences, but not excessive internalization in the corporate structure and activities, is vital to a company. Also, they found that via carefully selected channels, a firm's global commitment helps Japanese firms be more competitive and more likely to survive.

SA concepts and calculations were applied by Hough, Garitta and Sánchez (2004) to consumers' acceptance/rejection data of samples with different levels of sensory defects. The lognormal parametric model was found adequate for most defects and allowed prediction of concentration values corresponding to 10% probability of consumer rejection.

The state of the psychotherapy termination literature to date might best be characterized as inconclusive. Despite decades of studies, almost no predictors of premature termination have emerged consistently. An examination of this literature reveals a number of recurrent methodological-analytical problems that likely have contributed substantially to this state. SA, which was designed for longitudinal data on the occurrence of events, not only circumvents these problems but also capitalizes on the rich features of termination data and opens brand new avenues of investigation (Corning and Malofeeva. 2004).

From measurement of relationship between income inequality and the time-dependent risk (hazard) of a subsequent pregnancy (Gold et al., 2004) to self-reported teenagers crash involvements and citations (McCartt, 2003), to investigation of the role of product features in preventing customer churn (Larivière & Poel, 2004), to the improvement of operations management process in the provision of service (Pagell, 2004), and factors affecting corporate survival rates (Parker, 2002), additional applicability of SA in the business setting was significant and varied from personnel management to accounting to equipment maintenance and repair.

Combination with Other Methods

SA can be combined with many other decision models in the real world. Each model has its share of advantages and shortcomings. The complimentary effects, supporting arguments and the different view points may strengthen the final results.

A *feedforward neural network* architecture aimed at survival probability estimation is presented by Eleuteri, A. et al. (2003) which generalizes the standard, usually linear, models described in literature. The network builds an approximation to the survival probability of a system at a given time, conditional on the system features. The resulting model is described in a hierarchical *Bayesian* framework. Experiments with synthetic and real world data compare the performance of this model with the commonly used standard ones.

With the introduction of compulsory long-term care (LTC) insurance in Germany in 1995, a large claims portfolio with a significant proportion of censored observations became available. Czado and Rudolph (2002) presented an analysis of part of this portfolio using the *Cox proportional hazard* model to estimate transition intensities. In contrast to the more commonly used *Poisson regression* with graduation approach, where censored observations and time dependent risk factors are ignored, this approach allows the inclusion of both censored observations as well as time dependent risk factors such as time spent in LTC. Furthermore, they calculated premiums for LTC insurance plans in a multiple state *Markov process* based on these estimated transition intensities.

Vance and Geoghegan (2002) of US EPA National Center for Environmental Economics took as its point of departure a simple *utility-maximizing* model that suggests many possible determinants of deforestation in an economic environment characterized by missing or thin markets. Hypotheses from the model are tested on a data set that combines a time series of satellite imagery with data collected from a survey of farm households whose agricultural plots were geo-referenced using a *Global Positioning System* (GPS). Model results suggest that the deforestation process is characterized by non-linear duration dependence, with the probability of forest clearance first decreasing and then increasing with the passage of time.

Theoretical Improvements

Molinaro et al. (2004) proposed a unified strategy for estimator construction, selection, and performance assessment in the presence of censoring. A number of common estimation procedures follow this approach in the full data situation, but depart from it when faced with the obstacle of evaluating the loss function for censored observations. They argue that one can, and should, also adhere to this estimation road map in censored data situations.

While traditionally, SA included all of the information on a subject during a particular interval, period analyses look at just at survival experience in a recent time interval. Therefore it allows the researcher to limit or cut off the "survival experience" at the beginning and end of any chose interval, and allows it to be adapted to studies where short term survival is common. The idea is that therefore the results are less biased as Smith et al. (2004) proved. It is possible that this technique will be more widely used in the future as it seems to be more practical.

Multivariate survival data arises when subjects in the same group are related to each other or when there are multiple recurrences of the disease in the same subject. A common goal of SA is to relate the outcome (time to event) to a set of covariates. Gao, Manatunga and Chen (2004) focused on prognostic classification for multivariate survival data where identifying subgroups of patients with similar prognosis is of interest. They proposed a computationally feasible method to identify prognostic groups with the widely used Classification and Regression Trees (CART) algorithm, a popular one in data mining.

Limitations of SA

Underlying assumptions in the models, dealing with censored data and statistical power have been problems in this area. According to Fiddell & Tabachnick (2001, p.805), the challenging issues in SA "include testing the assumption of proportionality of hazards, dealing with censored data, assessing strength of association of models and individual covariates, choosing among the variety of statistical tests for differences among treatment groups and contributions of covariates, and interpreting odds ratios". Missing data is a common problem with SA. Larger samples are required for testing with covariates. Normality of sampling distributions, linearity, and homoscedasticity can lead to the results in better increased predictability, and less difficulty dealing with outliers. Censored cases should be systematically similar to those remaining in the study; otherwise the selection can no longer be considered randomly assigned. The conditions ought to remain constant throughout the experiment. Those assumptions are challengeable.

As with any scientific method there is an element of art that needs to be added in order to make the theory more usable. Certainly, SA is subject to GIGO (Garbage In, Garbage Out) because the results of a SA can be strongly influenced by the presence of error in the original data. As with any form of data gathering and analysis it is important that the researchers use only information that can be considered as relevant to the subject at hand.

FUTURE TRENDS

The past centuries have shown great strides in the development of the field of SA and there is no reason for their use to become anything but more important. As the computer age continues and advanced mathematical problems are able to be solved with the stroke of a few

keys, the use of SA will only become more important and play a greater role in our everyday lives.

Certainly, using incorrect models will lead to erroneous results/conclusions. We can imagine in the future, a single and unified model may dramatically increase the power for all SA studies. Also, SDM as a new branch of data mining may integrate with other data mining tools.

SA is based on a foundation of common principles and a common goal, there is no end in sight to transformations of SA methodologies, and there are constantly new variations on the theme and new applications for those variations. The use of SA is a significant contribution to society and will increase longevity of populations in the future.

CONCLUSION

Aside from mathematics and economics, SA is mostly used in the medical field. Gradually, SA has also been widely used in the social sciences where interest is on analyzing time to events such as job changes, marriage, birth of children and so forth. To the extent that the second paradigm of Mantel began a mere 50 years ago, the expansion and development of SA today is indeed remarkable. The progress of Kaplan-Meier, Mantel, Cox, Aalen as well as others not even mentioned here have proven SA is a reliable scientific tool susceptible to the rigors of modern mathematics. In order to properly administer treatment, caregivers and pharmaceutical providers should incorporate SA into the decision-making process. The same holds true for the effective management of business operations. It demonstrates that SA or SDM is a dynamic field, with many advances since its inception, as well as many opportunities for evolving in the future.

Technology and SA must simply enjoy a symbiotic relationship for both to flourish. SA is a dynamic and developing science that has no boundaries, other than those which are imposed upon it by human limitations.

REFERENCES

Bland, M. & Douglas, A. (1998, Dec.). Statistics Notes: Survival Probabilities – the Kaplan-Meier method. *British Medical Journal.*

Blandón, J.G. (2001). The Timing of Foreign Direct Investment under Uncertainty: Evidence from the Spanish Banking Sector, *Journal of Economic Behavior & Organization, 45(2), 213-224.*

Corning, A.F. and Malofeeva, E.V. (2004). The Application of Survival Analysis to the Study of Psychotherapy Termination, *Journal of Counseling Psychology, 51(3), 354-367.*

Czado, C. and Rudolph, F. (2002). Application of Survival Analysis Methods to Long-term Care Insurance**,** *Insurance: Mathematics and Economics, 31(3), 395-413.*

Eleuteri, A. et al. (2003). A Novel Neural Network-based Survival Analysis Model, *Neural Networks, 16(5-6), 855-864.*

Fiddell, L. & Tabachnick, B. (2001). *Using Multivariate Statistics. Massachusetts*. Allyn & Bacon.

Gao, F., Manatunga, A.K and Chen S. (2004). Identification of Prognostic Factors with Multivariate Survival Data, *Computational Statistics & Data Analysis, 45(4), 813-824.*

Gold et al. (2004). Income Inequality and Pregnancy Spacing, *Social Science & Medicine, 59(6), 1117-1126.*

Gudmundsson, S.V. and Rhoades, D.L. (2001). Airline Alliance Survival Analysis: Typology, Strategy and Duration, *Transport Policy, 8(3), 209-218.*

Harrington, D (2003). History of Survival Analysis. The Internet <http://filebox.vt.edu/org/stathouse/Survival.html>

Holford, T. (2002). *Multivariate Methods in Epidemiology*. New York: Oxford University Press, Inc.

Hough G., Garitta L. and Sánchez R. (2004). Determination of Consumer Acceptance Limits to Sensory

Defects Using Survival Analysis, *Food Quality and Preference, Available online.*

Kimura, F. and Fujii, T. (2003). Globalizing Activities and the Rate of Survival: Panel Data Analysis on Japanese Firms, *Journal of the Japanese and International Economies, 17(4), 538-560.*

Larivière, B. and Poel, D. V. (2004). Investigating the Role of Product Features in Preventing Customer Churn, by using Survival Analysis and Choice Modeling: The Case of Financial Services, *Expert Systems with Applications, 27(2), 277-285.*

Leung, K., Elashoff, R., Afifi, A. (1997). Censoring Issues in Survival Analysis. *Annual Review of Public Health,* 18, 83-104.

Lo, A.W, MacKinlay, A.C. and Zhang, J (2002). Econometric Models of Limit-order Executions, *Journal of Financial Economics, 65(1), 31-71.*

McCartt, A.T. et al. (2003). Driving Experience, Crashes and Traffic Citations of Teenage Beginning Drivers, *Accident Analysis & Prevention, 35(3), 311-320.*

Molinaro, A.M. et al. (2004). Tree-based Multivariate Regression and Density Estimation with Right-censored Data, *J. of Multivariate Analysis, 90(1), 154-177.*

Morriso, J. (2004) Introduction to Survival Analysis in Business. *The Journal of Business Forecasting.*

NCSS (2003). NCSS 2004 Statistical Analysis System. The Internet <http://www.ncss.com/ncsswin.html>

Oakes, D. (2000, March). Survival Analysis, *J. of the American Statistical Association,* 282-285.

Pagell, M. & Melnyk, S. (2004), Assessing the Impact of Alternative Manufacturing Layouts in a Service Setting, *Journal of Operations Management,* 22, 413-429.

Pallant, J. (2007). SPSS survival manual: A step by step guide to data analysis using SPSS. Auckland, New Zealand:Allen & Unwind.

Parker et al. (2002). Corporate Governance and Corporate Failure: a Survival Analysis, Corporate Governance, 2(2), 4-12.

SAS (2004). Survival Data Mining: Predictive Hazard Modeling for Customer History Data. The Internet http://support.sas.com/training/us/crs/bmce.html

SAS (2003). SAS/STAT. The Internet http://www.sas.com/technologies/analytics/statistics/stat/

Smith, L. et al. (2004) Providing More Up-to-date Estimates of Patient Survival: a Comparison of Standard Survival Analysis with Period Analysis Using Life-table Methods and Proportional Hazards Models. *Journal of Clinical Epidemiology,* 57(1), 14-20.

Statsoft, Inc. (2003). Survival/Failure Time Analysis, The Internet http://www.stasoftinc.com/textbook/stsurvan.html

Tableman, Mara (2003). *Survival Analysis Using S: Analysis of Time-to-Event Data*. Chapman & Hall/CRC.

Teradata (2003). New Customer Survival Analysis Solution for Telcos. The Internet http://www.businesswire.com

Vance C. and Geoghegan J. (2002). Temporal and Spatial Modeling of Tropical Deforestation: a Survival Analysis Linking Satellite and Household Survey Data**,** *Agricultural Economics, 27(3), 317-332.*

KEY TERMS

Censored: Censored cases are those in which the survival times are unknown.

Cumulative Proportion Surviving: This is the cumulative proportion of cases surviving up to the respective interval. Since the probabilities of survival are assumed to be independent across the intervals, this probability is computed by multiplying out the probabilities of survival across all previous intervals. The resulting function is also called the survivorship or survival function.

Failure Analysis: computing the time it takes for a manufactured component to fail.

Hazard Function: A time to failure function that gives the instantaneous probability of the event (failure) given that it has not yet occurred.

Life Tables: describing the survival rate as a function of time, referred to as the survivor function.

Lifetime (or failure time, survival data): Data that measure lifetime or the length of time until the occurrence of an event.

Proportion Failing: This proportion is computed as the ratio of the number of cases failing in the respective interval, divided by the number of cases at risk in the interval.

Survival Time: The time to the occurrence of a given event.

Symbiotic Data Miner

Kuriakose Athappilly
Western Michigan University, USA

Alan Rea
Western Michigan University, USA

INTRODUCTION

Symbiotic data mining is an evolutionary approach to how organizations analyze, interpret, and create new knowledge from large pools of data. Symbiotic data miners are trained business and technical professionals skilled in applying complex data-mining techniques and business intelligence tools to challenges in a dynamic business environment.

BACKGROUND

Most experts agree (Piatetsky-Shapiro, 2000; Thearling, 2007) that data mining began in the 1960s with the advent of computers that could store and process large amounts of data. In the 1980s, data mining became more common and widespread with the distribution of relational databases and SQL. In the 1990s, business saw a boom in data mining as desktop computers and powerful server-class computers became affordable and powerful enough to process large amounts of data in data warehouses (Havenstein, 2007) as well as real-time data via online analytical processing (OLAP). Today we see an increasing use of advanced processing of data with the help of artificial intelligence technology tools such as fuzzy logic, decision trees, neural networks, and genetic algorithms (Brachman, et al., 1996; Gargano & Raggad, 1999). Moreover, current trends are moving organizations to reclassify data mining as business intelligence using such tools as Cognos (Cognos, 2007).

We also see three distinct theoretical approaches to data mining: statistical (classical), Artificial Intelligence (heuristics), and machine learning (blended AI and statistics). The three approaches do not adhere to the historical boundaries applied to data mining; rather, they are embarkation points for data-mining practitioners (Thuraisingham, 1999; Kudyba & Hoptroff, 2001; Kepner & Kim, 2003; Padmanabhan & Tuzhilin, 2003). It is not the intent of this discussion to argue which approach best informs data mining. Instead, we note that many software platforms adhere to one or more methods for solving problems via data-mining tools.

Most organizations agree that sifting through data to create business intelligence, which they can use to gain a competitive edge, is an essential business component (Lee & Siau, 2001; Brown, 2004; MacInnis, 2004; Burns, 2005). Whether it is to gain customers, increase productivity, or improve business processes, data mining can provide valuable information if it is done correctly. In most cases, a triad of business manager, information technology technician, and statistician is needed to even begin the data-mining process. Although this combination can prove useful if a symbiotic relationship is fostered, typically the participants cannot effectively work with one another because they do not speak the same language. The manager is concerned with the business process, the technician with software and hardware performance, and the statistician with analyses of data and interpretations of newfound knowledge. While this may be an overgeneralization, it is not far from the truth (O'Hara, 2007).

What is needed, then, is an individual who can pull all three components together: a symbiotic data miner trained in business, technology, and statistics.

MAIN FOCUS

In this chapter we will discuss how an individual trained not only in business but also in technology and statistics can add value to any data mining and business intelligence effort by assisting an organization to choose the right data-mining techniques and software, as well as interpret the results within an informed business context.

Data Mining in Contemporary Organizations

Data mining is the "semi-automatic discovery of patterns, associations, changes, anomalies, rules, and statistically significant structures and events in data" (Dhond, et al., 2000, p. 480). Analyzed data is many times larger than the sum of its parts. In other words, data mining can find new knowledge from observing relationships among the attributes in the form of predictions, clustering, or associations that many experts might miss. The new knowledge in a continuously changing environment is the most potent weapon for organizations to become and remain competitive (Inmon, 1996; Amato-McCoy, 2006; Corbitt, 2006).

In today's business organizations intelligence is necessary to anticipate economic trends, predict potential revenue streams, and create processes to maximize profits and efficiency. This is especially true for strategic and other mid-level managers (Athappilly, 2003). In the past, many decisions were made using corporate experience and knowledge experts. This is still true today. However, with the increased influx of data—some experts argue that the amount of information in the world doubles every 20 months (Dhond, et al., 2000; Tallon & Scannell, 2007)—many high-level managers now turn to data-mining software in order to more effectively interpret trends and relationships among variables of interest (Deal, 2004).

To support data mining an increasing amount of funds are invested in complex software to glean the data for patterns of information; hardware is purchased that can effectively run the software and distribute the results, and personnel are continually retrained or hired. The personnel include IT technicians, knowledge experts, statisticians, and various business managers. The mix of personnel needed to effectively collect, glean, analyze, interpret, and then apply data-mined knowledge ultimately can lead to one of the biggest data-mining challenges: communicating results to business managers so that they can make informed decisions. Although the managers are the ones who ultimately make the decisions, they do not have the necessary skills, knowledge base, and techniques to assess whether the heuristics, software, and interpreted results accurately inform their decisions. There is ultimately a disjunction between theoretical interpretation and pragmatic application (Athappilly, 2004).

The Challenge for Contemporary Organizations

The challenge is two-fold: 1. A shortcoming of many data-mining tools is the inability of anyone except experts to interpret the results. Business managers must be able to analyze the results of a data-mining operation to "help them gain insights . . . to make critical business decisions" (Apte, et al., 2002, p. 49) and 2. Business managers must rely on IT technicians to apply rules and algorithms, and then rely on statisticians and other experts to develop models and to interpret the results before applying them to a business decision. This process adds at least two layers between the decision and the data. Moreover, there are numerous opportunities for miscommunication and misinterpretation among the team members (O'Hara, 2007).

In order to flatten the layers between the requisite gleaned knowledge and its interpretation and application, a new type of business IT professional is needed to create a symbiotic relationship, which can sustain itself without the triadic team member requirements and the inherent polarities among them.

The Solution for Contemporary Organizations

The solution to the complex data-mining process is symbiotic data miners. The symbiotic data miner is a trained business information system professional with a background in statistics and logic. A symbiotic data miner not only can choose the correct data-mining software packages and approaches but also analyze and glean knowledge from large data warehouses. Combined with today's complex analysis and visualization software, such as Clementine (SPSS, 2007) and Enterprise Miner (SAS, 2007), the symbiotic data miner can create illustrative visual displays of data patterns and apply them to specific business challenges and predictions.

Just as today's business managers use spreadsheets to predict market trends, analyze business profits, or manage strategic planning, the symbiotic data miner can fulfill the same functions on a larger scale using complex data-mining software. Moreover, the miner can also directly apply these results to organizational missions and goals or advise management on how to apply the gleaned knowledge.

Figure 1 demonstrates how the symbiotic data miner (Athappilly, 2002) is situated at the crux of data-mining technology, statistics, logic, and application. These components of business (corporate needs, actionable decisions, and environmental changes), technology (relational databases, AI, and interactive tools), and statistical and theoretical models (math/stat and visualization tools) all flow into the symbiotic data miner's immediate working space of Model Base software, such as SAS Enterprise Miner (SAS, 2007) and SPSS Clementine (SPSS, 2007). Interacting directly with the model-based application software, the symbiotic miner develops reliable models for explorative, as well as predictive purposes. After obtaining reliable models, the symbiotic data miner provides business intelligence through multilayer charts, tables, and graphs via dashboards or other devices to inform business executives making actionable decisions.

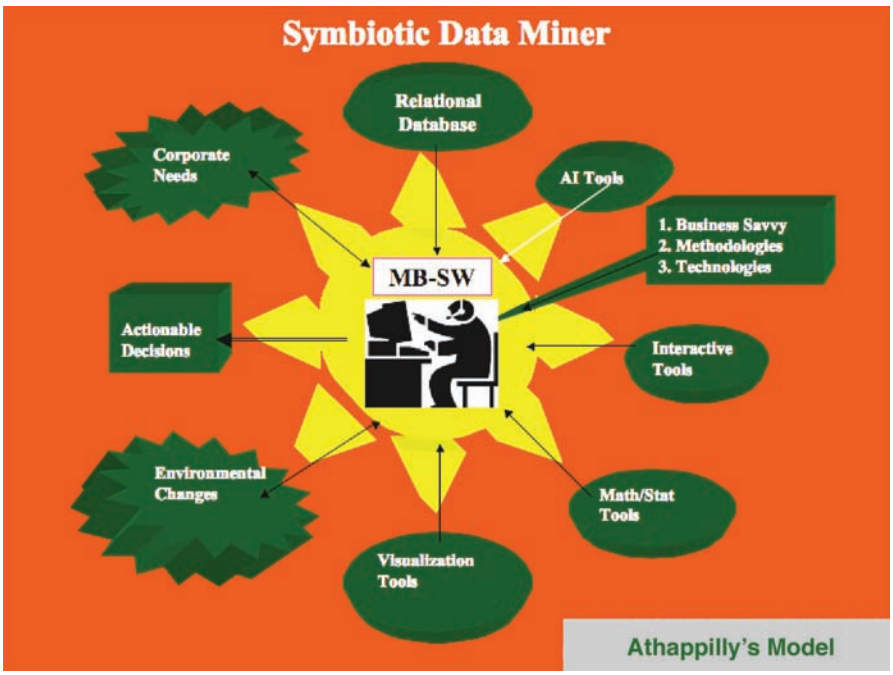
Ideal model-based application software should also have a built-in expert system capability. Using this capability, the symbiotic data miner can also provide recommendations after assessing several models. Thus, having these two sources of information—the business intelligence and recommendations—readily available from the symbiotic miner, executives can make informed decisions efficiently and effectively, thereby gaining a competitive advantage in today's highly competitive global business environment.

The symbiotic data miner plays a crucial role in flattening the layers between data-mining theory and statistics, technical support, and business acumen. The symbiotic miner can reduce miscommunication and bring applicable knowledge to a business challenge more quickly than a triadic team of business manager, technician, and statistician. Businesses should not replace all managers, technician, and statisticians with miners. However, organizations must infuse their data-mining decisions and business intelligence departments with symbiotic data miners to bridge the chasm between theory and practice.

FUTURE TRENDS

In the near future, organizations will have allocated positions for symbiotic data miners. Whatever their nomenclature, these business-informed technologically adept individuals will play a crucial role in strategic management decisions and long-term mission planning initiatives. The fledging business intelligence departments of today will continue to grow and integrate themselves into every aspect of the organizational

Figure 1. Symbiotic data miner



structure. Through sheer success, these departments will be subsumed into every department with symbiotic data miners specializing in particular business aspects.

Even further into the future, symbiotic data mining will simply become a way of doing business. No longer will the distinction among manager, technician, and statistician be as distinguishable. As software infused with business protocols and data-mining technology increases (Rea & Athappilly, 2004), business will implement data-mining systems on the desktop. Business managers at all levels will use symbiotic data-mining software as easily as many use office suite software (e.g., Microsoft Office) today. Software infused and linked with various data-mining systems and data sources will generate complex visual graphs, tables, and diagrams that can be mined according to one's desired results. We are already seeing the kernels of interconnected symbiotic mining in such applications as Microsoft Dynamics (Microsoft, 2007) and SAP's Business Warehouses (SAP, 2007).

The software that supports data mining will be user friendly, transparent and intuitive. Simultaneously, users will have experienced increased exposure to higher education, become more familiar with quantitative methods and technology tools, and will be better informed of the business culture and environment. As a result, data mining will be an inevitable routine activity implemented to make more informed decisions.

The catalysts for this movement will be a collective force comprised of educators, students, software vendors, and business professionals. Through internships, continued research, and increased business and academic partnerships and collaborations, the integration of business, data-mining technology, statistics, and theory into practical business software will become a reality. The realm of the symbiotic data miner will become a shared discipline in the information-rich business environment.

CONCLUSION

Symbiotic data miners are trained in business, information technology, and statistics. They are able to implement data mining solutions, interpret the results, and then apply them to business challenges. Without people with these skills, businesses will not be able to take full advantage of the vast stores of data available to them in order to make informed decisions.

However, the symbiotic data miner will not come about without changes in how individuals are trained in higher education and on the job. Without a combination of business, information technology, statistics, and logic we cannot look for an infusion of symbiotic data miners anytime soon. Today, businesses have begun to realize the need to tap into the vast knowledge stores available to them. As organizations move more toward business intelligence we will witness the emergence of more symbiotic data miners even though we may not identify them by this name.

REFERENCES

Amato-McCoy, D. (2006). The Power of Knowledge. *Chain Store Age*, 82(6), 48.

Apte, C., Liu, B., Pednault, E., & Smyth, P. (2002). Business Applications of Data Mining. *Communications of the ACM*, 45(8), 49-53.

Athappilly, K. (2002). *Symbiotic Mining: An Antidote for Corporate Insanity*. Paper presented at the meeting of High Performance Computing (HiPC), Bangalore, India. (http://www.hipc.org/hipc2002/2002Posters/SymbMining.doc)

Athappilly, K. (2003). Data Mining Coming of Age and Corporate Insanity in Diminishing Returns. *The 39th Annual Meeting of Midwest Business Administration Association Proceedings*, Chicago, IL.

Athappilly. K. (2004). Data Mining at Crossroads: A Retailer's Story. *Proceedings of the 40th Annual Meeting of Midwest Business Administration Association*. Chicago, IL, 75-82.

Brachman, R., Khabaza, T., Kloesgen, W., Piatetsky-Shapiro, G., & Simoudis, E. (1996). Mining Business Databases. *Communications of the ACM*, 39(11), 42-48.

Brown, J. (2004). IT Users Seek to Close Performance Management Gap. *Computing Canada*, 30(7), 10.

Burns, M. (2005). Business Intelligence Survey. *CA Magazine*, 138(5), 18.

Cognos. (2007). *Enterprise Business Intelligence*. Retrieved May 4, 2007 from http://www.cognos.com/products/businessintelligence/

Corbitt, T. (2006). The Power of Data: Mining and Warehousing. *Credit Management*, April, 32-33.

Deal, K. (2004). The Quest for Prediction. *Marketing Research*, 16(4), 45-47.

Dhond, A., Gupta, A., Vadhavkar, S. (2000). Data Mining Techniques for Optimizing Inventories for Electronic Commerce. *Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Boston, MA, 480-486.

Gargano, M. & Raggad B. (1999). Data Mining – A Powerful Information Creating Tool. *OCLC Systems and Services*, 15(2), 81-90.

Havenstein, H. (2007). IT Opens Data Warehouses to External Users. *Computerworld*, 41(36), 16, 20.

Inmon, W.H. (1996). The Data Warehouse and Data Mining. *Communications of the ACM*, 39(11), 49-50.

Kepner, J. & Kim, R. (2003). Cluster Detection in Databases: The Adaptive Matched Filter Algorithm and Implementation. *Data Mining and Knowledge Discovery*, 7(1), 57-79.

Kudyba, S. & Hoptroff, R. (2001). *Data Mining and Business Intelligence: A Guide to Productivity*. Hershey, PA: Idea Group Publishing.

Lee, S. & Siau, K. (2001). A Review of Data Mining Techniques. *Industrial Management & Data Systems*, 101(1), 41-46.

MacInnis, P. (2004). Bullish on Business Intelligence. *Computing Canada*, 30(9), 20.

Microsoft. (2007). *Microsoft Dynamics*. Retrieved June 10, 2007 from http://www.microsoft.com/dynamics/gp/default.mspx

O'Hara, M. (2007). Strangers in a Strange Land: Knowing, Learning and Education for the Global Knowledge Society. *Futures*, 39(8), 930.

Padmanabhan, B. & Tuzhilin, A. (2003). On the Use of Optimization for Data Mining: Theoretical Interactions and eCRM Opportunities. *Management Science*, 49(10), 1327-1343.

Piatetsky-Shapiro, G. (2000). Knowledge Discovery in Databases: 10 Years After. *SIGKDD Explor. Newsl.*, 1(2), 59-61.

Rea, A. & Athappilly, K. (2004). End-User Data Mining Using The E2DM Prototype: A Discussion of Prototype Development, Testing, and Evaluation. *Proceedings of the 2004 Midwest Decision Sciences Institute*, Cleveland, OH.

SAP. (2007). *SAP Business Warehouse*. Retrieved June 10, 2007 from http://help.sap.com/saphelp_nw04/helpdata/en/b2/e50138fede083de10000009b38f8cf/content.htm

SAS. (2007). *Enterprise Miner*. Retrieved June 3, 2007 from http://www.sas.com/technologies/analytics/datamining/miner/

SPSS. (2007). *Clementine*. Retrieved June 3, 2007 from http://www.spss.com/clementine/

Tallon, P. & Scannell, R. (2007). Information Life Cycle Management. *Communications of the ACM*, 50(11), 65-69.

Thearling, K. (2007). *An Introduction to Data Mining*. Retrieved May 28, 2007 from http://www.thearling.com/text/dmwhite/dmwhite.htm

Thuraisingham, B. (1999). *Data Mining: Technologies, Techniques, Tools, and Trends*. Boca Raton, FL: CRC Press.

KEY TERMS

Artificial Intelligence: A field of information technology that studies how to imbue computers with human characteristics and thought. Expert systems, natural language, and neural networks fall under the AI research area.

Business Intelligence: Information that enables high-level business managers and executives to make strategic and long-term business decisions.

Clementine: Data-mining software owned by SPSS Corporation that consists of supervised and unsupervised learning algorithms to create application models to solve business challenges.

Cognos: Business intelligence software that enables organizations to monitor performance and develop strategic business solutions based on collected data.

Decision Trees: Tree-shaped structures that represent sets of decisions. Different types of decisions trees, such as Classification and Regression Trees (CART), allow experts to create validated decision models that can then be applied to new data sets.

Enterprise Miner: Data-mining software developed by SAS Corporation that consists of supervised and unsupervised learning algorithms to create application models to solve business challenges.

Fuzzy Logic: A type of logic that does not rely on a binary yes or no. Instead computer systems are able to rank responses on a scale of 0.0 to 1.0 with 0.0 being false to 1.0 being true. This allows computer systems to deal with probabilities rather than absolutes.

Genetic Algorithms: A large collection of rules that represent all possible solutions to a problem. Inspired by Darwin's theory of evolution, these rules are simultaneously applied to data using powerful software on high-speed computers. The best solutions are then used to solve the problem.

Heuristics: A set of rules derived from years of experience to solve problems. These rules can be drawn from previous examples of business successes and failures. Artificial intelligence models rely on these rules to find relationships, patterns, or associations among variables.

Machine Learning: This involves a combination of AI and statistics. Software programs are able to predict and learn approaches to solve problems after repeated attempts.

Neural Networks: An artificial intelligence program that attempts to learn and make decisions much like the human brain. Neural networks function best with a large pool of data and examples from which they can learn.

Online Analytical Processing (OLAP): OLAP tools allow users to analyze different dimensions of multidimensional data.

Structured Query Language (SQL): This is a standardized query language used to pull information from a database.

Symbiotic Data Miner: An individual trained in business, information technology, and statistics. The symbiotic data miner is able to implement data mining solutions, interpret the results, and then apply them to business challenges.

Tabu Search for Variable Selection in Classification

Silvia Casado Yusta
Universidad de Burgos, Spain

Joaquín Pacheco Bonrostro
Universidad de Burgos, Spain

Laura Nuñez Letamendía
Instituto de Empresa, Spain

INTRODUCTION

Variable selection plays an important role in classification. Before beginning the design of a classification method, when many variables are involved, only those variables that are really required should be selected. There can be many reasons for selecting only a subset of the variables instead of the whole set of candidate variables (Reunanen, 2003): (1) It is cheaper to measure only a reduced set of variables, (2) Prediction accuracy may be improved through the exclusion of redundant and irrelevant variables, (3) The predictor to be built is usually simpler and potentially faster when fewer input variables are used and (4) Knowing which variables are relevant can give insight into the nature of the prediction problem and allows a better understanding of the final classification model. The importance of variables selection before using classification methods is also pointed out in recent works such as Cai et al.(2007) and Rao and Lakshminarayanan (2007).

The aim in the classification problem is to classify instances that are characterized by attributes or variables. Based on a set of examples (whose class is known) a set of rules is designed and generalised to classify the set of instances with the greatest precision possible. There are several methodologies for dealing with this problem: Classic Discriminant Analysis, Logistic Regression, Neural Networks, Decision Trees, Instance-Based Learning, etc. Linear Discriminant Analysis and Logistic Regression methods search for linear functions and then use them for classification purposes. They continue to be interesting methodologies.

In this work an "ad hoc" new method for variable selection in classification, specifically in discriminant analysis and logistic regression, is analysed. This new method is based on the metaheuristic strategy tabu search and yields better results than the classic methods (stepwise, backward and forward) used by statistical packages such as SPSS or BMDP, as it's shown below. This method is performed for 2 classes.

BACKGROUND

Research in variable selection started in the early 1960s (Lewis, 1962 and Sebestyen, 1962). Over the past four decades, extensive research into feature selection has been conducted. Much of the work is related to medicine and biology (e.g. Inza et al., 2002; Shy and Suganthan, 2003). The selection of the best subset of variables for building the predictor is not a trivial question, because the number of subsets to be considered grows exponentially with the number of candidate variables, which means that feature selection is a NP (Nondeterministic Polynomial) -Hard computational problem (see Cotta et al., 2004). When the size of the problem is large finding an optimum solution in practice is not feasible.

Two different methodological approaches have been developed for variable selection problems: optimal or exact techniques (enumerative techniques), which can guarantee an optimal solution, but are applicable only in small sets; and heuristic techniques, which can find good solutions (although they cannot guarantee the optimum) in a reasonable amount of time. Among the former, the Narendra and Fukunaga (1977) algorithm is one of the best known, but as Jain and Zongker (1997) pointed out, this algorithm is impractical for problems with very large feature sets. Recent references about implicit enumerative techniques of selection features adapted to regression models could be found in Gatu and

Kontoghiorghes (2005) and (2006). On the other hand, among the heuristic techniques we find works based on genetic algorithms (see Bala et al. (1996), Jourdan et. al. (2001)), Oliveira et al, 2003 and Wong and Nandi, (2004)) and the recent work by García et al. (2006) who present a method based on Scatter Search. As found in other optimization problems metaheuristic techniques have proved to be superior methodologies.

However, although there are many references in the literature regarding selecting variables for their use in classification, there are very few key references on the selection of variables for their use in discriminant analysis and logistic regression. For this specific purpose the *Stepwise* method (Efroymson, 1960*)* and all its variants, such as O'Gorman's (2004), as well as the *Backward* and *Forward* methods, can be found in the literature. These are simple selection procedures based on statistical criteria which have been incorporated into some of the best known statistical packages such as SPSS or BMDP. As highlighted by Huberty (1994) these methods are not very efficient, and when there are many original variables the optimum is rarely achieved.

SOLVING THE PROBLEM

Setting Out the Problem

We can formulate the problem of selecting the subset of variables with superior classification performance as follows: V being a set of m variables, such that $V = \{1, 2,..., m\}$ and A being a set of instances, (also named "training" set). For each case we also know the class it belongs to. Given a predefined value $p \in \mathrm{N}$, $p < m$, we have to find a subset $S \subset V$, with a size p with the greatest classification capacity, $f(S)$.

To be precise, for the discriminant analysis the function $f(S)$ is defined as a percentage of hits in A obtained through the variables of S with Fisher's classifier (Fisher, 1936).

For logistic regression the classification capacity, *f(s),* is obtained as follows: let's consider $S = \{1, 2, \ldots, p\}$ without loss of generality; let x_{ij} be the value of variable j for case i, $i \in A$, $j \in S$, and $y_i = 1$ if case i belongs first class and 0 otherwise; this classifier is obtained by calculating vector $\mathbf{c} = (c_0. c_1. \ldots. c_p)$ by maximizing the next expression

$$L(\mathbf{c}) = \prod_{i=1}^{n} p_i^{\,y_i} \cdot (1 - p_i)^{1-y_i}$$

where

$$p_i = \frac{1}{1 + e^{-z_i}}$$

and $z_i = c_o + c_1 x_{i1} + c_2 x_{i2} + \ldots + c_p x_{ip}$;

then, once vector **c** is obtained, every case i is classified to class 1 if $p_i > 0.5$ and in class 2 otherwise. Function L is usually named as likelihood function.

Solution Approach: Tabu Search

Tabu Search (TS) is a strategy proposed by Glover (1989 and 1990). A comprehensive tutorial on Tabu Search can be found in Glover and Laguna (2002).

Our Tabu Search algorithm includes a method for building an initial solution and a basic procedure for exploring the space of solutions around this initial solution The performance of the Tabu Algorithm is outlined as follows:

Tabu Search Procedure
Build an initial solution
Repeat
Execute Basic Tabu Search
until a stopping condition is reached

In this work a limit of 30 minutes of computational time is used as stopping condition. Next these elements are described.

Initial Solution

The initial solution is built as follows: starting from the empty initial solution, a variable is added in each iteration until the solution S reaches p variables ($|S| = p$). To decide which variable is added to the solution in each iteration the value of f is used.

Description of a Basic Algorithm

Our Tabu Search algorithm uses neighbouring moves which consist in exchanging an element that is in solu-

tion *S* for an outside element at each step. In order to avoid repetitive looping when a move is performed, consisting in exchanging *j* from *S* for *j'* from *V-S*, element *j* is prevented from returning to *S* for a certain number of iterations. We define *vector_tabu* (*j*) = the number of the iterations in which element *j* leaves *S*.

Some 'tabu' moves can be permitted under specific conditions ("aspiration criterion"), for example, to improve the best solution found. The basic Tabu Search method is described next, where *S* is the current solution and *S** the best solution. The *Tabu_Tenure* parameter indicates the number of iterations during which an element is not allowed to return to *S*. After different tests, *Tabu_Tenure* was set as *p*.

Basic Tabu Search Procedure
(a) *Read initial solution S*
(b) Do vector_tabu(j) = - Tabu_Tenure, j =1..m; niter =0, iter_better=0 and S = S*
(c) Repeat
(c.1) niter = niter+1
(c.2) Calculate $v_{jj'} = f(S \cup \{j'\} - \{j\})$
(c.3) Determine $v_{j^*j'^*} = max \{v_{jj'} / \forall\ j \in S, j' \notin S$ *verifying:*
niter > vector_tabu(j) + Tabu_Tenure or $v_{jj'} > f(S^*)$ *('aspiration criterion')}*
(c.4) Do $S = S \cup \{j'^*\} - \{j^*\}$ *and vector_tabu (j*) = niter*
(c.5) If f (S) > f(S) then do: S* = S, f* = f and iter_better = niter;*
until niter > iter_better+2·m

That is, this procedure terminates when $2 \cdot m$ iterations have taken place without improvement.

Computational Results

To check and compare the efficacy of these new method a series of experiments was run with different test problems. Specifically six data sets were used. These data sets can be found in the well-known data repository of the University of California, UCI, (see Murphi and Aha. 1994). This can be found at: www.ics.uci.edu/~mlearn/MLRepository.html. The following databases were used:

- *Spambase Database*: 57 variables, 2 classes and 4,601 cases.
- *Mushrooms Database*: 22 variables, 2 classes and 8,100 cases. The 22 nominal variables were transformed into 121 binary variables: 1 for each binary variable and 1 per possible answer for the remaining variables. *Covertype Database*: This is a forestry database, with 54 explanatory variables, 8 classes and more than 580,000 cases or instances. *Conect-4 Opening Database:* 42 nominal variables, 3 classes and 67,557 cases. The 42 nominal variables were transformed into 126 binary variables.
- *Waveform Database:* 40 variables with continuous values, 3 classes and 5,000 instances. We have considered the two first classes.
- *Nursery Database*: 8 nominal variables, 5 classes and 12,960 cases. The 8 nominal variables were transformed into 28 binary variables. The 5 classes are grouped together in two classes ("not_recom" and the rest).

From each database we have randomly selected 10 sets of 200 cases as test sets for evaluating the model with independent data .

Our Tabu Search is compared to the classic *Stepwise, Backward* and *Forward* procedures used in some well-known statistical software packages such as SPSS or BMDP.

The experiments were divided into two groups. The first group is devoted to compare our Tabu Search with the traditional *Stepwise, Forward* and *Backward* methods for discriminant analysis and the second group for logistic regression. All the experiments were done on a Pentium IV 2.4 GHz PC using the BORLAND DELPHI compiler (version 5.0).

We apply our Tabu Search algorithm to different training sets from the above mentioned databases and subsecuently we evaluate the models thus obtained with independent data (test sets).

Discriminant Analysis

Table 1 presents a summary of the solutions obtained with the test sets for each value of p considered (classification capacity in the intermediary steps). Specifically, the data we use are 10 test sets obtained from each database which have been described previously. The results of *Forward* method are omitted because they are the same as the ones obtained by *Stepwise* method.

Table 1. Comparison in test sets for discriminant analysis

Data	*M*	*p*	***Backward***	***Stepwise***	**Tabu S**
Spam	57	3	0.787	0,787	**0,804**
	57	4	0.812	0,812	**0,825**
	57	5	0.827	0,827	**0,844**
	57	6	0.827	0,827	**0,852**
	57	7	0.843	0,831	**0,883**
	57	8	0.854	0,850	**0,883**
Mushrooms	121	3	0.907	0,952	**0,976**
	121	4	0.906	0,950	**0,989**
	121	5	0.915	0,989	**1,000**
Cover	54	3	**0.740**	**0,740**	0,732
	54	4	0.741	**0,752**	0,730
	54	5	**0.754**	0,752	0,749
	54	6	**0.759**	0,749	0,746
	54	7	**0.758**	0,755	0,726
Connect *Cover* *Cover*	*126*	3	0.729	0.585	**0.743**
	126	4	0.729	0.671	**0.742**
	126	5	0.729	0.677	**0.747**
	126	6	0.720	0.659	**0.738**
	126	*7*	0.720	0.678	**0.745**
	126	*8*	0.720	0.684	**0.744**
	126	*9*	0.682	0.692	**0.744**
	126	*10*	0.682	0.714	**0.768**
	126	*11*	0.682	0.714	**0.741**
	126	*12*	0.606	**0.764**	0.759
Wave *Cover*	*40*	*3*	0.867	0.868	**0.882**
	40	*4*	0.886	0.880	**0.891**
	40	*5*	**0.899**	0.890	0.893
	40	*6*	**0.900**	**0.900**	0.899
	40	*7*	**0.901**	**0.901**	0.898
Nursery *Cover*	*26*	*3*	*0.673*	0.673	**1.000**
	26	*4*	*0.673*	0.669	**1.000**
	26	*5*	*0.673*	0.672	**1.000**
	26	*6*	*0.673*	0.672	**1.000**

For every case table 1 shows the mean values of *f*. In bold is pointed for every case the best result.

The following points can be made regarding Table 1:

- Our Tabu Search algorithm improves the solutions of the classic methods in most of the cases.
- The *Backward* method seems to work similar than the *Stepwise* and *Forward* methods.

Logistic Regression

In table 2 a summary of the solutions obtained by logistic regression with the test sets for each value of

Table 2. Comparison in test sets for logistic regression

Data	*M*	*p*	***Backward***	***Stepwise***	**Tabu S**
Spam	57	3	0.834	0.834	**0.867**
	57	4	0.839	0.839	**0.871**
	57	5	0.855	0.855	**0.877**
	57	6	0.857	0.857	**0.884**
	57	7	0.868	0.868	**0.888**
	57	8	0.879	0.879	**0.900**
Mushrooms	121	3	0.860	0.860	**0.982**
	121	4	0.828	0.828	**0.995**
	121	5	0.810	0.810	**1.000**
Cover	54	3	0.671	0.671	**0.749**
	54	4	0.740	0.735	**0.749**
	54	5	0.760	**0.764**	0.751
	54	6	**0.755**	0.750	0.747
	54	7	**0.761**	**0.761**	0.755
Connect *Cover* *Cover*	*126*	3	0.736	0.741	**0.746**
	126	4	0.737	**0.747**	**0.747**
	126	5	0.746	0.749	**0.753**
	126	6	0.742	0.757	**0.765**
	126	7	0.749	**0.769**	0.765
	126	*8*	0.745	0.766	**0.777**
	126	*9*	0.743	0.773	**0.785**
	126	*10*	0.740	0.776	**0.779**
	126	*11*	0.741	0.782	**0.786**
	126	*12*	0.742	0.773	**0.791**
Wave *Cover*	*40*	*3*	0.865	0.865	**0.868**
	40	*4*	0.752	0.752	**0.892**
	40	*5*	**0.899**	**0.899**	0.894
	40	*6*	**0.899**	**0.899**	0.898
	40	*7*	0.865	0.865	**0.903**
Nursery *Cover*	*26*	*3*	**1.000**	**1.000**	**1.000**
	26	*4*	**1.000**	**1.000**	**1.000**
	26	*5*	**1.000**	**1.000**	**1.000**
	26	*6*	**1.000**	**1.000**	**1.000**

p considered (classification capacity in the intermediary steps) is shown. Specifically, the data we use are 10 test sets obtained from each database previously described. As for the discriminant analysis the results of *Forward* method are omitted because they are the same as the ones obtained by *Stepwise* method. For every case table 2 shows the mean values of *f*. In bold is pointed for every case the best result.

The following points can be made regarding Table2:

- The *Backward* method seems to work similar to the *Stepwise* and *Forward* method.
- Our Tabu Search algorithm improves the solutions of the classic methods for most of the cases.

FUTURE TRENDS

This work proposes an "ad hoc" new method for variable selection in classification, in particular in discriminant analysis and logistic regression. This method is performed for 2 classes but in the future it could be adapted to higher number of classes.

Also, as a continuity of this work, another metaheuristic strategies for variable selection in logistic regression and discriminant analysis will be designed and implemented to check and compare their efficacy.

CONCLUSIONS

A Tabu Search method to select variables that are subsequently used in discriminant analysis and logistic regression models is proposed and analysed. Although there are many references in the literature regarding variable selection for their use in classification, there are very few key references on the selection of variables for their use in discriminant analysis and logistic regression. After performing some tests it is found that Tabu Search obtains better results than the *Stepwise*, *Backward* or *Forward* methods used by classic statistical packages such as SPSS or BMDP.

ACKNOWLEDGMENTS

Authors are grateful for financial support from the Spanish Ministry of Education and Science and FEDER Funds (National Plan of R&D - Projects SEJ2005-08923/ECON) and from Regional Government of "Castilla y León" ("Consejeria de Educación" – Project BU008A06).

REFERENCES

Bala J., Dejong K., Huang J., Vafaie H. and Wechsler H. (1996): Using Learning to Facilitate the Evolution of Features for Recognizing Visual Concepts, *Evolutionary Computation*, 4, 3, 297-311.

Cai D.M., Gokhale M. and Theiler J. (2007): Comparison of Feature Selection and Classification Algorithms in Identifying Malicious Executables. *Computational Statistics & Data Analysis*, 51(6), 3156-3172.

Cotta C., Sloper C. and Moscato P. (2004): Evolutionary Search of Thresholds for Robust Feature Set Selection: Application to the Analysis of Microarray Data, *Lecture Notes In Computer Science* 3005: 21-30.

Efroymson, M.A. (1960): Multiple Regression Analysis, Mathematical Methods for Digital Computers (Ralston, A. and Wilf, H.S., ed.) Vol.1. Wiley, New York.

Fisher R.A.(1936). The Use of Multiple Measures in Taxonomic Problems. *Annual Eugenics* 7, 179-188.

García F.C., García M., Melián B., Moreno J.A. and Moreno M. (2006): Solving Feature Selection Problem by a Parallel Scatter Search. In press in *European Journal of Operational Research.*

Gatu C. and Kontoghiorghes E.J. (2005): Efficient Strategies for Deriving the Subset {VAR} Models. *Computational Management Science*, 2 (4):253-278.

Gatu C. and Kontoghiorghes E.J. (2006): Branch-and-bound Algorithms for Computing the Best-Subset Regression Models. *Journal of Computational and Graphical Statistics*, 15 (1):139-156.

Glover F. and Laguna M. (2002): Tabu Search, in Handbook of Applied Optimization, P. M. Pardalos and M. G. C. Resende (Eds.), Oxford University Press, pp. 194-208.

Glover F. (1989): Tabu Search*:* Part I, *ORSA Journal on Computing*. Vol. 1, pp. 190-206.

Glover F. (1990): Tabu Search*:* Part II, *ORSA Journal on Computing*. Vol. 2, pp. 4-32..

Huberty C.J.(1994): Applied Discriminant Analysis. Wiley. Interscience.

Inza, I., Sierra B. and Blanco R. (2002): Gene selection by sequential search wrapper approaches in microarray cancer class prediction. *Journal of Intelligent & Fuzzy Systems*, 12 (1), 25--33.

Jain A. and Zongker D. (1997): Feature Selection: Evaluation, Application, and Small Sample Performance, IEEE Trans. *Pattern Analysis and Machine Intelligence*, vol. 19, no. 2, pp. 153-158.

Jourdan L., Dhaenens C. and Talbi E. (2001): A Genetic Algorithm for Feature Subset Selection in Data-Mining for Genetics, MIC 2001 Proceedings, 4th Metaheuristics Internationl Conference, 29-34.

Lewis P.M. (1962): The Characteristic Selection Problem in Recognition Systems, *IEEE Trans. Information Theory*, vol. 8: 171-178.

Murphy P. M. and Aha. D. W. (1994): UCI repository of Machine Learning. University of California, Department of Information and Computer Science, http://www.ics.uci.edu/~mlearn/MLRepository.html

Narendra P.M. and Fukunaga K. (1977): A Branch and Bound Algorithm for Feature Subset Selection, *IEEE Trans. Computers*, vol. 26, no. 9: 917-922.

O'Gorman T.W. (2004): Using adaptive Methods to Select Variables in Case-Control Studies, *Biometrical Journal* 46,5, pp.595-605.

Oliveira L.S., Sabourin R., Bortolozzi F., et al.(2003): A Methodology for Feature Selection Using Multiobjective Genetic Algorithms for Handwritten Digit String Recognition, *International Journal Of Pattern Recognition And Artificial Intelligence* 17 (6): 903-929.

Rao K.R. and Lakshminarayanan S. (2007): Partial Correlation Based Variable Selection Approach for Multivariate Data Classification Methods, *Chemometrics and Intelligent Laboratory Systems* 86 (1), 68-81.

Reunanen, J. (2003): Overfitting in making comparisons between variable selection methods. *Journal of Machine Learning Research,* 3 (7/8), 1371--1382.

Sebestyen G. (1962): Decision-Making Processes in *Pattern Recognition*. New York: MacMillan.

Shy S. and Suganthan P.N. (2003): Feature Analysis and Clasification of Protein Secondary Structure Data, *Lecture Notes in Computer Science* 2714: 1151-1158.

Wong M.L.D. and Nandi A.K. (2004): Automatic Digital Modulation Recognition Using Artificial Neural Network and Genetic Algorithm. *Signal Processing* 84 (2): 351-365.

KEY TERMS

Backward: A variable selection method which begins by selecting all the variables and then, at each step, the variable that contributes least to the prediction of group membership is eliminated. Thus, as the result, those variables that contribute the most to the discrimination between groups are obtained.

Classification Methods: Methods used for designing a set of rules which let us to classify a set of instances with the greatest precision possible. These methods are based on a set of examples whose class is known.

Forward: A variable selection method which begins by selecting the most discriminant variable according to some criterion. It continues by selecting the second most discriminant variable and so on. The algorithm stops when none of the non-selected variables discriminates in a significant way.

Metaheuristics: Metaheuristics are high level procedures that coordinate simple heuristics, such as local search, to find solutions that are of better quality that those found by the simple heuristics.

Stepwise: A variable selection procedure which introduces or eliminates variables at each step, depending on how significant their discriminating capacity is. It also allows for the possibility of changing decisions taken in previous steps, by eliminating from the selected set a variable introduced in a previous step of the algorithm or by selecting a previously eliminated variable.

Tabu Search (TS): Is a metaheuristic strategy based on the premise that problem solving, in order to qualify as intelligent, must incorporate adaptive memory and responsive exploration. TS explores the solution space beyond de local optimum. Once a local optimum is reached, upward moves and those worsening the solutions are allowed. Simultaneously, the last moves are marked as tabu during the following iterations to avoid cycling.

Variable Selection Problem: This problem consists in finding from a set of *m* variables a subset of them that can carry out the classification task in an optimum way.

Techniques for Weighted Clustering Ensembles

Carlotta Domeniconi
George Mason University, USA

Muna Al-Razgan
George Mason University, USA

INTRODUCTION

In an effort to achieve improved classifier accuracy, extensive research has been conducted in classifier ensembles. Very recently, cluster ensembles have emerged. It is well known that off-the-shelf clustering methods may discover different structures in a given set of data. This is because each clustering algorithm has its own bias resulting from the optimization of different criteria. Furthermore, there is no ground truth against which the clustering result can be validated. Thus, no cross-validation technique can be carried out to tune input parameters involved in the clustering process. As a consequence, the user is not equipped with any guidelines for choosing the proper clustering method for a given dataset.

Cluster ensembles offer a solution to challenges inherent to clustering arising from its ill-posed nature. Cluster ensembles can provide more robust and stable solutions by leveraging the consensus across multiple clustering results, while averaging out emergent spurious structures that arise due to the various biases to which each participating algorithm is tuned.

In this chapter, we discuss the problem of combining multiple *weighted clusters*, discovered by a locally adaptive algorithm (Domeniconi, Papadopoulos, Gunopulos, & Ma, 2004) which detects clusters in different subspaces of the input space. We believe that our approach is the first attempt to design a cluster ensemble for subspace clustering (Al-Razgan & Domeniconi, 2006).

Recently, several subspace clustering methods have been proposed (Parsons, Haque, & Liu, 2004). They all attempt to dodge the curse of dimensionality which affects any algorithm in high dimensional spaces. In high dimensional spaces, it is highly likely that, for any given pair of points within the same cluster, there exist at least a few dimensions on which the points are far apart from each other. As a consequence, distance functions that equally use all input features may not be effective.

Furthermore, several clusters may exist in different subspaces comprised of different combinations of features. In many real-world problems, some points are correlated with respect to a given set of dimensions, while others are correlated with respect to different dimensions. Each dimension could be relevant to at least one of the clusters.

Global dimensionality reduction techniques are unable to capture local correlations of data. Thus, a proper feature selection procedure should operate locally in input space. Local feature selection allows one to embed different distance measures in different regions of the input space; such distance metrics reflect local correlations of data. In (Domeniconi, Papadopoulos, Gunopulos, & Ma, 2004) we proposed a *soft* feature selection procedure (called LAC) that assigns weights to features according to the local correlations of data along each dimension. Dimensions along which data are loosely correlated receive a small weight, which has the effect of elongating distances along that dimension. Features along which data are strongly correlated receive a large weight, which has the effect of constricting distances along that dimension. Thus the learned weights perform a directional local reshaping of distances which allows a better separation of clusters, and therefore the discovery of different patterns in different subspaces of the original input space.

The clustering result of LAC depends on two input parameters. The first one is common to all clustering algorithms: the number of clusters k to be discovered in the data. The second one (called h) controls the strength of the incentive to cluster on more features. The setting of h is particularly difficult, since no domain knowledge

for its tuning is likely to be available. Thus, it would be convenient if the clustering process automatically determined the relevant subspaces.

In this chapter we discuss two cluster ensemble techniques for the LAC algorithm. We focus on setting the parameter h and assume that the number of clusters k is fixed. We leverage the diversity of the clusterings produced by LAC when different values of h are used, in order to generate a consensus clustering that is superior to the participating ones.

BACKGROUND

In many domains it has been shown that a classifier ensemble is often more accurate than any of the single components. This result has recently initiated further investigation in ensemble methods for clustering. In (Fred & Jain, 2002) the authors combine different clusterings obtained via the *k*-means algorithm. The clusterings produced by *k*-means are mapped into a co-association matrix, which measures the similarity between the samples. Kuncheva et al. (Kuncheva & Hadjitodorov, 2004) extend the work in (Fred & Jain, 2002) by choosing at random the number of clusters for each ensemble member. The authors in (Zeng, Tang, Garcia-Frias, & Gao, 2002) introduce a meta-clustering procedure: first, each clustering is mapped into a distance matrix; second, the multiple distance matrices are combined, and a hierarchical clustering method is introduced to compute a consensus clustering. In (Hu, 2004) the authors propose a similar approach, where a graph-based partitioning algorithm is used to generate the combined clustering. Ayad et al. (Ayad & Kamel, 2003) propose a graph approach where data points correspond to vertices, and an edge exists between two vertices when the associated points share a specific number of nearest neighbors. In (Fern & Brodley, 2003) the authors combine random projection with a cluster ensemble. EM is used as clustering algorithm, and an agglomerative approach is utilized to produce the final clustering. Greene et al. (Greene, Tsymbal, Bolshakova, & Cunningham, 2004) apply an ensemble technique to medical diagnostic datasets. The authors focus on different generation and integration techniques for input clusterings to the ensemble. *K-means*, *K*-medoids and fast *weak clustering* are used as generation strategies. The diverse clusterings are aggregated into a co-occurrence matrix. Hierarchical schemes are then applied to compute the consensus clustering. Greene's approach follows closely Fred and Jain's approach (Fred & Jain, 2002). However, they differ in the generation strategies. Similarly, in (Boulis & Ostendorf, 2004) the association between different clusterings produced by various algorithms is investigated. Techniques based on constrained and unconstrained clustering and on SVD are considered. (Gionis, Mannila, & Tsaparas, 2005) approach finds an ensemble clustering that agrees as much as possible with the given clusterings. The proposed technique does not require the number of clusters as an input parameter, and handles missing data.

In (Strehl & Ghosh, 2003) the authors propose a consensus function aimed at maximizing the normalized mutual information of the combined clustering with the input ones. Three heuristics are introduced: Cluster-based Similarity Partitioning Algorithm (CSPA), HyperGraph Partitioning Algorithm (HGPA), and Meta-Clustering Algorithm (MCLA). All three algorithms transform the set of clusterings into a hypergraph representation.

In CSPA, a binary similarity matrix is constructed for each input clustering. An entry-wise average of all the matrices gives an overall similarity matrix S. S is utilized to recluster the data using a graph-partitioning based approach. HGPA constructs a hypergraph in which each hyperedge represents a cluster of an input clustering. The algorithm seeks a partitioning of the hypergraph by cutting a minimal number of hyperedges. The partition gives k unconnected components of approximately the same size. MCLA is based on the clustering of clusters. It provides object-wise confidence estimates of cluster membership. Hyperedges are grouped, and each data point is assigned to the collapsed hyperedge in which is participates most strongly.

MAIN FOCUS

In the following we introduce two consensus functions to identify an emergent clustering that arises from multiple clustering results. We reduce the problem of defining a consensus function to a graph partitioning problem (Dhillon, 2001; Fern & Brodley, 2004; Strehl & Ghosh, 2003). In fact, the *weighted clusters* computed by the LAC algorithm offer a natural way to define a similarity measure to be integrated as weights associated to

the edges of a graph. The overall clustering ensemble process is illustrated in Figure 1.

Locally Adaptive Clustering (LAC)

We briefly describe our locally adaptive clustering algorithm (Domeniconi, Papadopoulos, Gunopulos, & Ma, 2004). Consider a set of points in some space of dimensionality D. A *weighted cluster* C is a subset of data points, together with a vector of weights $\mathbf{w} = (w_1, ..., w_o)$, such that the points in C are closely clustered according to the L_2 norm distance weighted using $\mathbf{w}$. The component w_j measures the degree of correlation of points in C along feature j.

Our approach progressively improves the quality of initial centroids and weights, by investigating the space near the centers to estimate the dimensions that matter the most. We start with *well-scattered* points in a dataset S as the k centroids: we choose the first centroid at random, and select the others so that they are far from one another, and from the first chosen center. We initially set all weights to $1/D$. Given the initial centroids c_j, for $j = 1, ..., k$, we compute the corresponding sets

$$S_j = \left\{ x \mid \left(\sum_{i=1}^{D} w_{ji}(x_i - c_{ji})^2\right)^{1/2} < \left(\sum_{i=1}^{D} w_{li}(x_i - c_{li})^2\right)^{1/2}, \forall l \neq j \right\}$$

where w_{ji} and c_{ji} represent the ith components of vectors $\mathbf{w}_j$ and $\mathbf{c}_j$ respectively (ties are broken randomly). We then compute the average distance along each dimension from the points in S_j to $\mathbf{c}_j$:

$$X_{ji} = \frac{1}{|S_j|} \sum_{x \in S_j} (c_{ji} - x_i)^2,$$

where $|S_j|$ is the cardinality of set S_j. The smaller X_{ji} is, the larger is the correlation of points along dimension i. We use the value X_{ji} in an exponential weighting scheme to credit weights to features (and to clusters), as given in

$$w_{ji}^* = \frac{\exp(-X_{ji}/h)}{\sum_{i=1}^{D} \exp(-X_{ji}/h)}$$

(the parameter h controls the strength of the incentive for clustering on more features). The computed weights are used to update the sets S_j, and therefore the centroids' coordinates as given in

$$c_{ji}^* = \frac{1}{|S_j|} \sum_{x \in S_j} x_i.$$

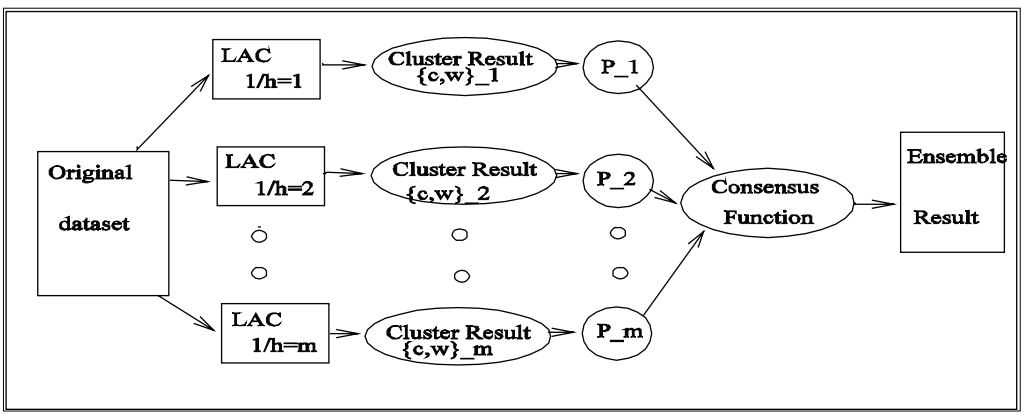
The procedure is iterated until convergence is reached. The resulting algorithm is called LAC.

Weighted Similarity Partitioning Algorithm (WSPA)

LAC outputs a partition of the data, identified by the two sets $\{\mathbf{c}_1,...,\mathbf{c}_k\}$ and $\{\mathbf{w}_1,...,\mathbf{w}_k\}$.

Our aim here is to generate robust and stable solutions via a consensus clustering method. We can generate contributing clusterings by changing the parameter h (as illustrated in Figure 1).

Figure 1. The clustering ensemble process



The objective is then to find a consensus partition from the output partitions of the contributing clusterings, so that an ``improved'' overall clustering of the data is obtained. The details of our approach are as follows**.**

For each data point $\mathbf{x}_i$, the weighted distance from cluster C_l is given by

$$d_{il} = \sqrt{\sum_{s=1}^{D} w_{ls}(x_{is} - c_{ls})^2}.$$

Let $D_i = \max_l\{d_{il}\}$ be the largest distance of $\mathbf{x}_i$ from any cluster. We want to define the probability associated with cluster C_l given that we have observed $\mathbf{x}_i$. At a given point $\mathbf{x}_i$, the cluster label C_l is assumed to be a random variable from a distribution with probabilities $\{P(C_l \mid \mathbf{x}_i)\}_{l=1}^{k}$. We provide a nonparametric estimation of such probabilities based on the data and on the clustering result. We do not make any assumption about the specific form of the underlying data distributions, thereby avoiding parameter estimations of models.

In order to embed the clustering result in our probability estimations, the smaller the distance d_{il} is, the larger the corresponding probability credited to C_l should be. Thus, we can define $P(C_l \mid \mathbf{x}_i)$ as follows:

$$P(C_l \mid \mathbf{x}_i) = \frac{D_i - d_{il} + 1}{kD_i + k - \sum_l d_{il}} \tag{1}$$

where the denominator serves as a normalization factor to guarantee $\sum_{l=1}^{k} P(C_l \mid \mathbf{x}_i) = 1$. We observe that $\forall l = 1, ..., k$ and $\forall i = 1, ..., n$ $P(C_l \mid \mathbf{x}_i) > 0$. In particular, the added value of 1 in (1) allows for a non-zero probability $P(C_L \mid \mathbf{x}_i)$ when $L = \arg\max_l\{d_{il}\}$.In the last case $P(C_l \mid \mathbf{x}_i)$ assumes its minimum value

$$P(C_L \mid \mathbf{x}_i) = \frac{1}{(kD_i + k + \sum_l d_{il})}.$$

For smaller distance values d_{il}, $P(C_l \mid \mathbf{x}_i)$ increases proportionally to the difference $D_i - d_{il}$: the larger the deviation of d_{il} from D_i, the larger the increase. As a consequence, the corresponding cluster C_l becomes more likely, as it is reasonable to expect based on the information provided by the clustering process. Thus, equation (1) provides a nonparametric estimation of the posterior probability associated to each cluster C_l.

We can now construct the vector P_i of posterior probabilities associated with $\mathbf{x}_i$:

$$P_i = (P(C_1 \mid \mathbf{x}_i), P(C_2 \mid \mathbf{x}_i), ..., P(C_k \mid \mathbf{x}_i))^t \tag{2}$$

where *t* denotes the transpose of a vector. The transformation $\mathbf{x}_i \rightarrow P_i$ maps the *D* dimensional data points $\mathbf{x}_i$ onto a new space of *relative coordinates* with respect to cluster centroids, where each dimension corresponds to one cluster. This new representation embeds information from both the original input data and the clustering result.

We then define the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$ as the cosine similarity between the corresponding probability vectors:

$$s(\mathbf{x}_i, \mathbf{x}_j) = \frac{P_i^t P_j}{\| P_i \| \| P_j \|} \tag{3}$$

We combine all pairwise similarities (3) into an $(n \times n)$ similarity matrix *S*, where $S_{ij} = s(\mathbf{x}_i, \mathbf{x}_j)$. We observe that each clustering may provide a different number of clusters, with different sizes and boundaries. The size of the similarity matrix *S* is independent of the clustering approach, thus providing a way to align the different clustering results onto the same space, with no need to solve a label correspondence problem.

After running the LAC algorithm *m* times for different values of the *h* parameter, we obtain the *m* similarity matrices $S_1, S_2, ..., S_m$. The combined similarity matrix Ψ defines a *consensus function* that can guide the computation of a consensus partition:

$$\Psi = \frac{1}{m}\sum_{l=1}^{m} S_l \tag{4}$$

Ψ_{ij} reflects the average similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$ (through P_i and P_j) across the *m* contributing clusterings.

We now map the problem of finding a consensus partition to a graph partitioning problem. We construct a complete graph $G = (V, E)$, where $|V|=n$ and the vertex V_i identifies $\mathbf{x}_i$. The edge E_{ij} connecting the vertices V_i and V_j is assigned the weight value Ψ_{ij}. We run METIS (Kharypis & Kumar, 1995) on the resulting graph to compute a *k*-way partitioning of the *n* vertices that minimizes the edge weight-cut. This gives the consensus clustering we seek. The size of the resulting

graph partitioning problem is n^2. We call the resulting algorithm WSPA (Weighted Similarity Partitioning Algorithm).

Weighted Bipartite Partitioning Algorithm (WBPA)

Our second approach maps the problem of finding a consensus partition to a bipartite graph partitioning problem. This mapping was first introduced in (Fern & Brodley, 2004). In (Fern & Brodley, 2004), however, 0/1 weight values are used. Here we extend the range of weight values to [0, 1].

In this context, the graph models both instances and clusters, and the graph edges can only connect an instance vertex to a cluster vertex, thus forming a bipartite graph.

Suppose, again, that we run the LAC algorithm m times for different values of the h parameter. For each instance $\mathbf{x}_i$, and for each clustering $\nu = 1, ..., m$ we then can compute the vector of posterior probabilities P_i^ν, as defined in equations (1) and (2). Using the P vectors, we construct the following matrix A:

$$A = \begin{pmatrix} (P_1^1)^t (P_1^2)^t ... (P_1^m)^t \\ (P_2^1)^t (P_2^2)^t ... (P_2^m)^t \\ \vdots \\ (P_n^1)^t (P_n^1)^t ... (P_n^m)^t \end{pmatrix}$$

Note that the $(P_i^\nu)^t$s are row vectors (t denotes the transpose). The dimensionality of A is therefore $n \times km$, under the assumption that each of the m clusterings produces k clusters. (We observe that the definition of A can be easily generalized to the case where each clustering may discover a different number of clusters.)

Based on A we can now define a bipartite graph to which our consensus partition problem maps. Consider the graph $G = (V, E)$ with V and E constructed as follows. $V = V^C \cup V^I$, where V^C contains km vertices, each representing a cluster of the ensemble, and V^I contains n vertices, each representing an input data point. Thus $|V| = km + n$. The edge E_{ij} connecting the vertices V_i and V_j is assigned a weight value defined as follows. If the vertices V_i and V_j represent both clusters or both instances, then $E(i, j) = 0$; otherwise, if vertex V_i represents an instance $\mathbf{x}_i$ and vertex V_j represents a cluster C_j^ν (or vice versa) then the corresponding entry of E is $A(i, k(\nu - 1) + j)$.

Note that the dimensionality of E is $(km + n) \times (km + n)$, and E can be written as follows:

$$E = \begin{pmatrix} 0 & A^t \\ A & 0 \end{pmatrix}$$

A partition of the bipartite graph G partitions the cluster vertices and the instance vertices simultaneously. The partition of the instances can then be output as the final clustering. Due to the special structure of the graph G (sparse graph), the size of the resulting bipartite graph partitioning problem is kmn. Assuming that $(km) < n$, this complexity is much smaller than the size n^2 of WSPA.

We again run METIS on the resulting bipartite graph to compute a k-way partitioning that minimizes the edge weight-cut. We call the resulting algorithm WBPA (Weighted Bipartite Partitioning Algorithm).

FUTURE TRENDS

In our future work we will consider utilizing our consensus function as a similarity matrix for hierarchical and spectral clustering. This approach will eliminate the requirement for balanced clusters. We will extend our approach to be used with any subspace clustering technique. In addition, we aim at designing an ensemble that preserves a subspace clustering structure. One possibility is to leverage the weight vectors associated with the input clustering that shares the highest NMI with the clustering produced by the ensemble (this can be performed using the RAND statistic). Another possibility is to infer a set of dimensions for each cluster from the clustering result of the ensemble.

The diversity-accuracy requirements of the individual clusterings, in order for the ensemble to be effective, will be also investigated. It is expected that the accuracy of the ensemble improves when a larger number of input clusterings is given, provided that the contributing clusterings are diverse.

CONCLUSION

We have discussed the cluster ensemble learning paradigm and introduced two cluster ensemble techniques

for the LAC algorithm. Our algorithms leverage the diversity of the clusterings produced by LAC when different values of *h* are used, in order to generate a consensus clustering that is superior to the participating ones.

REFERENCES

Al-Razgan, M., & Domeniconi, C. (2006, April 20-22). Weighted Clustering Ensembles. *SIAM International Conference on Data Mining.* Bethesda, Maryland.

Ayad, H., & Kamel, M. (2003). *Finding Natural Clusters Using Multi-clusterer Combiner Based on Shared Nearest Neighbors* (2709 ed.).

Boulis, C., & Ostendorf, M. (2004). Combining multiple clustering systems. *8th European conference on Principles and Practice of Knowledge Discovery in Databases(PKDD), LNAI 3202.*

Dhillon, I. S. (2001). Co-clustering documents and words using bipartite spectral graph partitioning. *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining.* San Francisco, California.

Domeniconi, C., Papadopoulos, D., Gunopulos, D., & Ma, S. (2004, April 22-24). Subspace Clustering of High Dimensional Data. *SIAM International Conference on Data Mining.* Florida, USA.

Fern, X. Z., & Brodley, C. E. (2003). Random projection for high dimensional data clustering: A cluster ensemble approach. *International Conference on Machine Learning (ICML).*

Fern, X. Z., & Brodley, C. E. (2004). Solving cluster ensemble problems by bipartite graph partitioning. *Proceedings of the twenty-first international conference on Machine learning.* Banff, Alberta, Canada.

Fred, A., & Jain, A. (2002). Data Clustering Using Evidence Accumulation. *International Conference on Pattern Recognition (ICPR'02).*

Gionis, A., Mannila, H., & Tsaparas, P. (2005). Clustering Aggregation. *International Conference on Data Engineering (ICDE).*

Greene, D., Tsymbal, A., Bolshakova, N., & Cunningham, P. (2004). Ensemble Clustering in Medical Diagnostics. *IEEE Symposium on Computer-Based Medical System.*

Hu, X. (2004). Integration of Cluster Ensemble and Text Summarization for Gene Expression Analysis. *Fourth IEEE Symposium on Bioinformatics and Bioengineering (BIBE'04).*

Kharypis, G., & Kumar, V. (1995). Multilevel k-way partitioning scheme for irregular graphs. *Technical report, University of Minnesota Department of Computer Science and Army HPC Research Center*.

Kuncheva, L., & Hadjitodorov, S. (2004). Using Diversity in Cluster Ensembles. *International Conference on Systems, Man and Cybernetics.*

Parsons, L., Haque, E., & Liu, H. (2004). Subspace clustering for high dimensional data: A review. *SIGKDD Explor. Newsl., 6*(1), 90-95.

Strehl, A., & Ghosh, J. (2003). Cluster ensembles—Acknowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res., 3*, 583-617.

Zeng, Y., Tang, J., Garcia-Frias, J., & Gao, G. R. (2002). An Adaptive Meta-Clustering Approach: Combining the Information from Different Clustering Results. *Proceedings of the IEEE Computer Society Conference on Bioinformatics.*

KEY TERMS

Clustering: The process of grouping objects into subsets, such that those within each cluster are more closely related to one another than objects assigned to different clusters, according to a given similarity measure.

Clustering Ensembles: Learning paradigm that aggregates a collection of clustering components to produce a consensus partition that is more accurate than the individual clusterings.

Curse of Dimensionality: Phenomenon that refers to the fact that, in high-dimensional spaces, data become extremely sparse and are far apart from each other. As a result, the sample size required to perform an accurate prediction in problems with high dimensionality is usually beyond feasibility.

Subspace Clustering: Simultaneous clustering of both row and column sets in a data matrix.

K-Way Graph Partitioning: Given a weighted graph $G = (V, E)$, partitioning G into k parts means to find a partition of V into k disjoint clusters of vertices. The sum of the weights of the crossed edges is defined as the cut of the partition. The general goal of graph partitioning is to find a k-way partition that minimizes the cut.

Locally Adaptive Clustering (LAC): Algorithm that discovers clusters in subspaces, spanned by different combinations of dimensions, via local weighting of features.

Weighted Bipartite Partition Algorithm (WBPA): Maps the problem of finding a consensus partition to a bipartite graph partitioning problem. The graph models both data points and clusters as vertices, and an edge can only connect an instance vertex to a cluster vertex, thus forming a bipartite graph. The algorithm utilizes the weighted features provided by LAC to map each data point onto a new space of posterior probabilities where each dimension corresponds to a cluster. The resulting posterior probability $P(C_l|\mathbf{x}_i)$ is the weight associated with the edge connecting vertex C_l with vertex $\mathbf{x}_i$.

Weighted Graph: A weighted graph is represented by $G = (V, E)$, where V is a set of vertices and E is a nonnegative and symmetric $|V|\times|V|$ matrix that characterizes the similarity between each pair of vertices.

Weighted Similarity Partition Algorithm (WSPA): Defines a consensus function for an ensemble of clusterings by mapping the problem onto a k-way partitioning problem. The algorithm utilizes the weighted features provided by LAC to define pairwise similarity measures between data points. It then constructs a weighted graph in which vertices correspond to data points and edges embed similarities between the corresponding pairs of points.

Temporal Event Sequence Rule Mining

Sherri K. Harms
University of Nebraska at Kearney, USA

INTRODUCTION

The emergence of remote sensing, scientific simulation and other survey technologies has dramatically enhanced our capabilities to collect temporal data. However, the explosive growth in data makes the management, analysis, and use of data both difficult and expensive. Methods that characterize interesting or unusual patterns from the volumes of temporal data are needed (Roddick & Spiliopoulou, 2002; Han & Kamber, 2005).

The association rule mining methods described in this chapter provide the ability to find periodic occurrences of inter-sequential factors of interest, from groups of long, non-transactional temporal event sequences. Association rule mining is well-known to work well for problems related to the recognition of frequent patterns of data (Han & Kamber, 2005). Rules are relatively easy for humans to interpret and have a long history of use in artificial intelligence for representing knowledge learned from data.

BACKGROUND

A time series database contains sequences of values typically measured at equal time intervals. There are two main categories of temporal sequences: *transaction-based sequences* and *event sequences*. A transaction-based sequence includes an identifier such as a customer ID, and data mining revolves around finding patterns within transactions that have matching identifiers. An example pattern is "A customer who bought Intel stock is likely to buy Google stock later." The transaction has a definite boundary around known items of interest. There are many techniques that address these problems (Han & Kamber, 2005).

Data analysis on event sequences is enormously more complex than transactional data analysis. Event sequences are often long streams of data where interesting patterns occur either within the sequence or across multiple sequences. There are no inherently defined boundaries (or identifiers) around factors that might be of interest. Temporal event sequence algorithms must be able to compute inference from volumes of data, find the interesting events involved, and define the boundaries around them. An example pattern is "A La Niña weather pattern is likely to precede drought in the western United States". La Niña weather data is based on Pacific Ocean surface temperatures and atmospheric values, and drought data is based on precipitation data from weather stations throughout the western United States. The sheer number of possible combinations of interesting factors and relationships between them can easily overwhelm human analytical abilities. Often there is a delay between the occurrence of an event and its influence on dependent variables. These factors make finding interesting patterns difficult.

Many different methods have been applied to temporal event sequences. In statistics, event sequence data is often called a marked point process. However, traditional methods for analyzing marked point processes are ill suited for problems with long, non-transactional sequences with numerous event types (Mannila et al. 1997). The association rule mining methods described in this chapter extract meaningful inter-sequential patterns in this type of data. Additionally, the mined rules provide much richer information than correlation coefficients from correlating entire sequences. The methods described in this chapter are similar to inductive logic programming, but with an emphasis on time-limited occurrences of sequential data. Similarities also exist to algorithms used in string matching and bioinformatics, but the classes of patterns differ (Mannila et al. 1997).

MAIN FOCUS

Mining association rules is typically decomposed into three sub-problems: 1) prepare the data for analysis, 2) find frequent patterns and 3) generate association rules from the sets representing those frequent patterns (Agrawal et al., 1993).

Preparing Event Sequences for Analysis

To prepare sequential data for association rule mining, the data is discretized and partitioned into sequences of events. Typically, the data is normalized and segmented into partitions that have similar characteristics within a given interval. Each partition identifier is called an event type. Different partitioning methods and interval sizes produce diverse discretized versions of the same dataset. Proper discretization relies on domain-expert involvement. When multivariate sequences are used, each is normalized and discretized independently.

Partitioning methods include symbolizing (Lin et al., 2003) and intervals (Hoppner, 2002). The assignment of values to a certain state is somewhat arbitrary near the decision boundaries. Mörchen & Ultsch (2005) presented a method for meaningful unsupervised discretization that reduces the vulnerability to outliers in the data and reduces the problems that occur when intervals are cut in high density regions of data values.

Mielikäinen et al. (2006) proposed a discretization technique that segments data by aggregating the results of several segmentation algorithms and choosing the discretization that agrees as much as possible with the underlying structure of the data.

Finding Frequent Episodes based on Sliding Window Technologies

A discretized version of the time series is referred to as an *event sequence*. An event sequence $\hat{S}$ is a finite, time-ordered sequence of events (Mannila et al., 1995). That is, $\hat{S} = (e_1, e_2, \ldots e_n)$. An event is an occurrence of an event type at a given timestamp. The time that a given event e_i occurs is denoted i, and $i \leq i+1$ for all i timestamps in the event sequence. A sequence includes events from a single finite set of event types. An event type can be repeated multiple times in a sequence. For example, the event sequence $\hat{S}_1 = AABCAB$ is a sequence of 6 events, from a set of 3 event types {A,B,C}. In this event sequence, an A event occurs at time 1, followed by another A event, followed by a B event, and so on. The step size between events is constant for a given sequence.

An *episode* in an event sequence is a combination of events with partially specified order (Mannila et al., 1997). It occurs in a sequence if there are occurrences of events in an order consistent with the given order, within a given time bound. Formally, an episode α is a pair *(V, ordering)*, where *V* is a collection of events and the ordering is *parallel* if no order is specified, and *serial* if the events of the episode have fixed order. The episode length is defined as the number of events in the episode.

Finding frequent episodes in sequences was first described by Mannila et al. (1995). Frequent episodes are discovered by using a sliding window approach, *WINEPI*. A *window* on an event sequence $\hat{S}$ is an event subsequence, $w = e_i, e_{i+1}, \ldots e_{i+d}$ where the width of window *w*, denoted *d*, is the time interval of interest. The set of all windows *w* on $\hat{S}$, with a width of *d* is denoted $\hat{W}(\hat{S},d)$. In this system, the value of the window width is user-specified, varying the closeness of event occurrences. To process data, the algorithm sequentially slides the window of width *d* one step at a time through the data. The *frequency* of an episode α is defined as the fraction of windows in which the episode occurs. For example, in the sequence $\hat{S}_1$ above, if a sliding window of width 3 is used, serial episode α = AB occurs in the first window (AAB), the second window (ABC), and the fourth window (CAB). The guiding principle of the algorithm lies in the "downward-closed" property of frequency, which means every subepisode is at least as frequent as its superepisode. Candidate episodes with *(k+1)* events are generated by joining frequent episodes that have *k* events in common, and episodes that do not meet a user-specified frequency threshold are pruned.

The closure principle was first applied to the event sequences by Harms et al. (2001) to use only a subset of frequent episodes, called frequent closed episodes. A closed sequential pattern is a sequential pattern which has no super-sequence with the same occurrence frequency. A frequent closed episode is the intersection of all frequent episodes containing it. For example, in the $\hat{S}1$ sequence, using a window width d = 3, and a minimum frequency of three windows, serial episode α = AB is a frequent closed episode since no larger frequent episode contains it, and it meets the minimum frequency threshold. Using closed episodes results in a reduced input size and in a faster generation of the episodal association rules, especially when events occur in clusters. Casas-Garriga (2005) added post-processing of closed sequences to generate classical partial orders, without dealing directly with input data.

Hoppner & Klawonn (2002) divided multivariate sequences into small segments and discretized them based on their qualitative descriptions (such as increas-

ing, high value, convexly decreasing, etc.). Patterns are discovered in the interval sequences based on Allen's (1983) temporal interval logic. For example, pattern "A meets B" occurs if interval A terminates at the same point in time at which B starts. For any pair of intervals there is a set of 13 possible relationships: after, before, meets, is-met-by, starts, is-started-by, finishes, is-finished-by, overlaps, is-overlapped-by, during, contains and equals. This approach also finds frequent patterns by using sliding windows and creating a set of candidate *(k+1)*-patterns from the set of frequent patterns of size *k*. Moerchen (2006) enhanced this line of research by presenting a hierarchical language for expressing concise patterns that represent the underlying temporal phenomena, using coincidence and partial orders. Winarko & Roddick (2007) also used Allen's temporal relations to discover interval-based relationships.

Ng & Fu (2003) mined frequent episodes using a tree-based approach for event sequences. The process is comprised of two phases: 1) tree construction and 2) mining frequent episodes. Each node in the tree is labeled by an event, and also contains a count and a node type bit. First, the frequencies of each event are gathered and sorted by descending frequencies. The tree is built similar to the FP-Growth method (Han et al., 2000) but uses sliding windows rather than transactions.

An approach that finds patterns related to a user-specified target event type is introduced in (Sun et al., 2003). Because a sliding window approach may exclude useful patterns that lie across a window boundary, this approach moves the window to the next event of interest. That is, a window always either starts from or ends with a target event. Interesting patterns are those that frequently occur together with the target event and are relatively infrequent in the absence of the target event.

Generating Rules based on Sliding Window Technologies

As introduced by Mannila et al. (1995), association rules are generated in a straightforward manner from the frequent episodes. An episodal association rule *r* is a rule of the form $X \Rightarrow Y$, where X is *antecedent* episode, Y is the *consequent* episode, and $X \cap Y = \varnothing$. For sliding window approaches used on a set of event sequences T, the *support* of rule $X \Rightarrow Y$ is the percentage of windows in T that contain $X \cup Y$. The rule holds in T with *confidence c* if *c%* of the windows in T that contain X also contain Y.

Harms et al. (2001) used *representative episodal association rules*, to reduce the number of rules while maintaining rules of interest to the user. A set of representative episodal association rules is a minimal set of rules from which all rules can be generated. Usually, the number of representative episodal association rules is much smaller than the total number of rules.

Other Event Sequence Rule Mining Technologies

MINEPI, an approach that uses minimal occurrence of episodes rather than a sliding window was developed by Mannila et al. (1997). A *minimal occurrence* of an episode α in an event sequence $\hat{S}$, is a window $w=[t_s, t_e]$, such that 1) α occurs in the window w, 2) α does not occur in any proper subwindow of w, and 3) the width of window w is less than the user-specified maximum window width parameter. In this definition, timestamp t_s records the starting time of the occurrence of the episode, and t_e records it's ending time, and $t_s \leq t_e$. The width of window w equals $t_e - t_s + 1$. The minimal occurrence window widths are not constant for a given episode, but are the minimal amount of elapsed time between the start of the episode occurrence and the end of the episode occurrence. The support of an episode α is the number of minimal occurrences of α in $\hat{S}$. An episode α is considered frequent if its support conforms to the given minimum support threshold.

Meger & Rigotti (2004) presented *WinMiner*, an algorithm based on *MINEPI*, which extracts episode rules satisfying frequency, confidence and maximum gap constraints. They also find the smallest window size that corresponds to a local maximum of confidence for the rule (i.e., confidence is locally lower, for smaller and larger windows).

Harms & Deogun (2004) introduced *MOWCATL*, which finds inter-sequential patterns, with respect to user-specified constraints. The method finds rules of the form $\alpha_{[wina]} \Rightarrow_{lag} \beta_{[winc]}$, where the antecedent episode α occurs within a given maximum antecedent window width win_a, the consequent episode β occurs within a given maximum consequent window width win_c, and the start of the consequent follows the start of the antecedent within a given time *lag*. The rule confidence is the conditional probability that β occurs, given that

α occurs, under the time constraints specified by the rule. The support of the rule is the number of times the rule holds in the dataset. The *MOWCATL* algorithm first stores the occurrences of the event types (single event episodes) that meet the user-specified inclusion constraints. Larger episodes are built from smaller episodes by joining episodes with overlapping minimal occurrences, which occur within the maximum window width. After finding the supported episodes for the antecedent and the consequent independently, they are combined to form episodal association rules, where the start of the consequent follows the start of the antecedent within a *lag* in time between the occurrences of the antecedent and the respective occurrences of the consequent.

Laxman et al. (2007) presented frequent episode discovery under the notion of non-overlapped occurrences of episodes within an event sequence. Two occurrences of an episode are said to be non-overlapped if no event corresponding to one occurrence appears between events corresponding to the other. The corresponding episode frequency is defined as the cardinality of the largest set of non-overlapped occurrences of the episode in the given event sequence.

Gwadera et al. (2005) presented an approach to detect suspicious subsequences with low false alarm rates that depends on the probabilistic characteristics of the event stream. The most likely number of occurrences of an episode is calculated, as well as the probability of deviating from it. From this, a threshold is derived and used to detect suspicious subsequences. With this approach, the probability of missing real unusual activities is small.

FUTURE TRENDS

Not only are the rules generated by the above techniques useful by themselves, they could be analyzed in more detail with other methods, such as marked point process, rule induction or classification trees to provide global models of the data (Mannila, et al., 1997).

Often, temporal sequences have a spatial component. For future work, the methods described here should be expanded to consider the spatial extent of the relationships. Additionally, the rule discovery process will need to spatially interpolate areas that do not have observed data.

Another problem with most temporal data is that it occurs in the form of data streams, which are potentially unbounded in size. Research issues in data stream mining are presented in Gabor et al. (2005) and Jiang & Gruenwald (2006). Materializing all data is unrealistic and expensive if it could be stored; techniques that retrieve approximate information are needed. Additionally, parallel and distributed algorithms will be needed to handle the volume of data.

CONCLUSION

Various techniques have been applied to the three association mining sub-problems used to find inter-sequential patterns within non-transactional event sequences. Partitioning and discretization methods prepare the data for analysis while closely matching the underlying structure of the data. Window-based methods with closure or temporal relations are used to find frequent patterns and generate rules. Alternative approaches use minimal occurrences to discover time-lagged relationships between multiple event sequences or use probabilistic characteristics of the event sequence to detect suspicious subsequences.

However, analysis techniques that capture the global model of the data and incorporate the spatial or streaming component of temporal event sequences are in their infancy. This research area is a rich and largely unexplored field.

REFERENCES

Agrawal R., Faloutsos, C., & Swami, A. (1993). Efficient similarity search in sequence databases. *Proc. 4th International Conference on Foundations of Data Organizations and Algorithms* (pp. 69-84). Chicago, IL.

Allen, J.F. (1983). Maintaining knowledge about temporal intervals. *Communications of the ACM*. 26(110), 832-843.

Casas-Garriga, G. (2005). Summarizing sequential data with closed partial orders. In *Proc. SIAM International Data Mining Conference* (pp. 380–391). Newport Beach, CA.

Gabor, M.M., Zaslavsky, A., & Krishnaswamy, S. (2006). Mining Data Streams: A Review. *SIGMOD Record*. 34(2), 18-26.

Gwadera, R., Atallah, M. & Szpankowski, W. (2005). Reliable Detection of Episodes in Event Sequences, *Knowledge and Information Systems*, 7(4), 415-437.

Han, J., Pei, J., & Yin, Y. (2000). Mining Frequent Patterns without Candidate Generation. *Proc. 2000 SIGMOD*. Dallas, TX.

Han, J., Kamber, M. (2005). *Data Mining: Concepts and Techniques*. 2nd Edition. San Francisco, CA: Morgan Kaufmann.

Harms, S. K., Saquer, J., Deogun, J., & Tadesse, T. (2001). Discovering Representative Episodal Association Rules from Event Sequences Using Frequent Closed Episode Sets and Event Constraints, *Proc. ICDM '01* (pp. 603-606). Silicon Valley, CA.

Harms, S. K., & Deogun, J. (2004). Sequential Association Rule Mining with Time Lags, *Journal of Intelligent Information Systems (JIIS)*. 22 (1), 7-22.

Hoppner, F., & Klawonn F. (2002). Finding informative rules in interval sequences. *Intelligent Data Analysis*, 237-256.

Jiang, N. & Gruenwald, L. (2006). CFI-Stream: mining closed frequent itemsets in data streams. *Proc. KDD-06*. Philadelphia, PA.

Lin, J., Keogh, E., Lonardi, S., & Chiu, B. (2003). A Symbolic Representation of Time Series, with Implications for Streaming Algorithms. *Proc. 8th ACM SIGMOD Workshop on Research Issues in DMKD* (pp. 2-11). San Diego, CA.

Laxman, S., Sastry, P.S., & Unnikrishnan, K. P. (2007). A fast algorithm for finding frequent episodes in event streams. In *Proc. KDD-07* (San Jose, California).

Mannila, H., Toivonen, H. & Verkamo, A.I. (1995). Discovering frequent episodes in sequences. In U.Fayyad, M. R. Uthurusamy (Eds.) *Proc. KDD-95* (pp. 210-215). Montreal, Quebec, Canada.

Mannila, H., Toivonen, H., & Verkamo, A. I. (1997). Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*. 1 (3), 259-289.

Méger, N. & Rigotti, C. (2004). Constraint-based mining of episode rules and optimal window sizes. In *Proc. 8th European Conference on Principles and Practice of Knowledge Discovery in Databases* (Pisa, Italy). J. Boulicaut, et al., Eds. Lecture Notes in Computer Science, vol. 3202. Springer-Verlag New York, New York, NY, 313-324.

Mielikäinen, T., Terzi, E., & Tsaparas, P. (2006). Aggregating time partitions. In *Proc. KDD-06* (pp.347-356). Philadelphia, PA.

Mörchen, F. & Ultsch, A. (2005). Optimizing time series discretization for knowledge discovery. In *Proc. KDD-05* (pp. 660-665). Chicago, Illinois.

Mörchen, F. (2006). Algorithms for Time Series Knowledge Mining. In *Proc. KDD-06* (pp. 668-673). Philadelphia, PA.

Ng, A., & Fu, A.W. (2003). Mining Frequent Episodes for relating Financial Events and Stock Trends. *Proc. PAKDD 2003*. Seoul, Korea.

Roddick, J.F., & Spilopoulou, M. (2002). A Survey of Temporal Knowledge Discovery Paradigms and Methods. *Transactions on Data Engineering*. 14 (4), 750-767.

Sun, X., Orlowska, M.E., & Zhou, X. (2003). Finding Event-Oriented Patterns in Long Temporal Sequences. *Proc. PAKDD 2003*. Seoul, Korea.

Winarko, E. & Roddick, J.F. (2007). ARMADA - An Algorithm for Discovering Richer Relative Temporal Association Rules from Interval-based Data. *Data and Knowledge Engineering* 63(1), 76-90.

KEY TERMS

Episode: A combination of events with a partially specified order. The episode ordering is *parallel* if no order is specified, and *serial* if the events of the episode have a fixed order.

Episodal Association Rule: A rule of the form $X \Rightarrow Y$, where X is antecedent episode, Y is the consequent episode and $X \cap Y = \varnothing$. The confidence of an episodal association rule is the conditional probability that the consequent episode occurs, given the antecedent episode occurs, under the time constraints specified. The support of the rule is the number of times it holds in the database.

Event: An occurrence of an event type at a given timestamp.

Event Sequence: A finite, time-ordered sequence of events. A sequence of events $\hat{S}$ includes events from a single finite set of event types.

Event Type: A discretized partition identifier that indicates a unique item of interest in the database. The domain of event types is a finite set of discrete values.

Minimal Occurrence: A minimal occurrence of an episode α in an event sequence $\hat{S}$, is a window $w=[t_s, t_e]$, such that 1) α occurs in the window w, 2) α does not occur in any proper subwindow of w, and 3) the width of window w is less than the user-specified maximum window width parameter. Timestamps t_s and t_e records the starting and ending time of the episode, respectively, and $t_s \leq t_e$.

Window: An event subsequence, e_i, e_{i+1}, …e_{i+d} in a event sequence, where the width of the window, denoted d, is the time interval of interest. In algorithms that use sliding windows, the frequency of an episode is defined as the fraction of windows in which the episode occurs.

Temporal Extension for a Conceptual Multidimensional Model

Elzbieta Malinowski
Universidad de Costa Rica, Costa Rica

Esteban Zimányi
Université Libre de Bruxelles, Belgium

INTRODUCTION

Data warehouses integrate data from different source systems to support the decision process of users at different management levels. Data warehouses rely on a multidimensional view of data usually represented as relational tables with structures called *star* or *snowflake schemas*. These consist of *fact tables*, which link to other relations called *dimension tables*. A fact table represents the focus of analysis (e.g., analysis of sales) and typically includes attributes called *measures*. Measures are usually numeric values (e.g., quantity) used for performing quantitative evaluation of different aspects in an organization. Measures can be analyzed according to different analysis criteria or *dimensions* (e.g., store dimension). Dimensions may include *hierarchies* (e.g., month-year in the time dimension) for analyzing measures at different levels of detail. This analysis can be done using on-line analytical processing (OLAP) systems, which allow dynamic data manipulations and aggregations. For example, the roll-up operation transforms detailed measures into aggregated data (e.g., daily into monthly or yearly sales) while the drill-down operations does the contrary.

Multidimensional models include a time dimension indicating the timeframe for measures, e.g., 100 units of a product were sold in March 2007. However, the time dimension cannot be used to keep track of changes in other dimensions, e.g., when a product changes its ingredients. In many cases the changes of dimension data and the time when they have occurred are important for analysis purposes. Kimball and Ross (2002) proposed several implementation solutions for this problem in the context of relational databases, the so-called *slowly-changing dimensions.* Nevertheless, these solutions are not satisfactory since either they do not preserve the entire history of data or are difficult to implement. Further, they do not consider the research realized in the field of temporal databases.

Temporal databases are databases that support some aspects of time (Jensen & Snodgrass, 2000). This support is provided by means of different temporality types[1], to which we refer in the next section. However, even though temporal databases allow to represent and to manage time-varying information, they do not provide facilities for supporting decision-making process when aggregations of high volumes of historical data are required. Therefore, a new field called *temporal data warehouses* joins the research achievements of temporal databases and data warehouses in order to manage time-varying multidimensional data.

BACKGROUND

Temporal support in data warehouses is based on the different temporality types used in temporal databases. *Valid time* (VT) specifies the time when data is true in the modeled reality, e.g., the time when a specific salary was paid for an employee. Valid time is typically provided by users. *Transaction time* (TT) indicates the time when data is current in the database and may be retrieved. It is generated by the database management system (DBMS). Both temporality types, i.e., valid time and transaction time, can be combined defining *bitemporal time* (BT). Finally, changes in time defined for an object as a whole define the *lifespan* (LS) of an object, e.g., the time when an employee was working for a company.

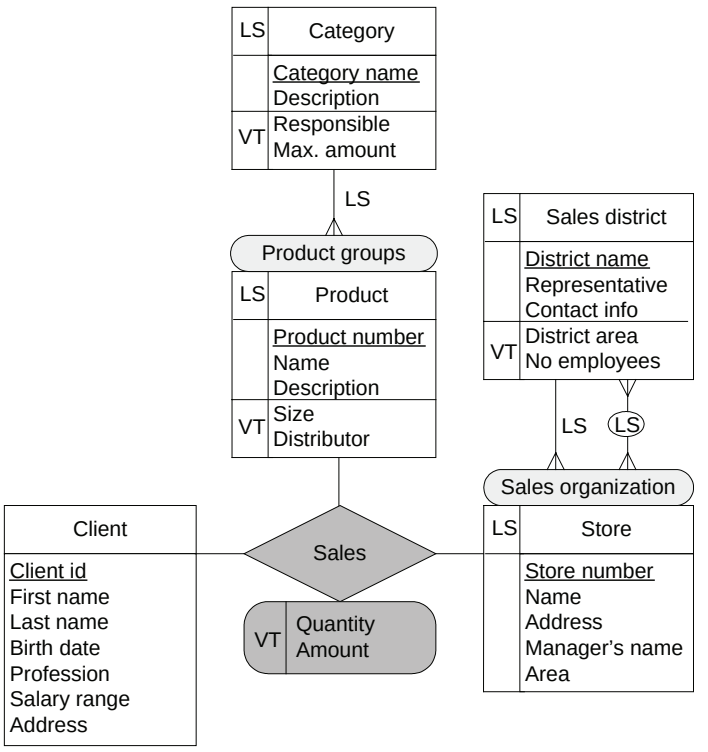
One characteristic of temporality types is their precision or *granularity*, indicating the duration of the time units that are relevant for an application. For example, the valid time associated to an employee's salary may be of granularity month. On the other hand, transaction time being system defined is typically of a millisecond granularity.

There is still lack of an analysis determining which temporal support is important for data warehouse ap-

plications. Most works consider valid time (e.g., Body, Miquel, Bédard, & Tchounikine, 2003; Wrembel & Bebel, 2007; Mendelzon & Vaisman, 2003). To our knowledge, no work includes lifespan support in temporal data warehouses. However, lifespan is important since it can help to discover, e.g., how the exclusion of some products influences sales. On the other hand, very few works relate to transaction time. For example, Martín and Abelló (2003) transform transaction time from source systems to represent valid time. This approach is semantically incorrect because data may be included in databases after their period of validity has expired. Further, transaction time coming from source system plays an important role in temporal data warehouses when traceability is required, e.g., for fraud detection. Other authors consider transaction time generated in temporal data warehouses in the same way as transaction time in temporal databases (e.g., Martín & Abelló, 2003; Mendelzon & Vaisman, 2003; Ravat & Teste, 2006). However, since data in temporal data warehouses is neither modified nor deleted, transaction time in a data warehouse represents the time when data was loaded into a data warehouse. Therefore, we propose to call it *loading time* (LT) (Malinowski & Zimányi, 2006a). LT can differ from transaction time or valid time of source systems due to the delay between the time when the changes have occurred in source systems and the time when these changes are integrated into a temporal data warehouse. Another approach (Brucker & Tjoa, 2002) considers valid time, transaction time, and loading time. However, they limit the usefulness of these temporality types for only active data warehouses, i.e., for data warehouses that include event-condition-action rules (or triggers).

The inclusion of temporal support raise many issues, such as efficient temporal aggregation of multidimensional data (Moon, Vega, & Immanuel, 2003), correct aggregation in presence of data and schema changes (Body *et al.*, 2003; Eder, Koncilia, & Morzy, 2002; Wrembel & Bebel, 2007; Mendelzon & Vaisman, 2003; Golfarelli, Lechtenbörger, Rizzi, & Vossen, 2006), or temporal view materialization from non-temporal sources (Yang & Widom, 1998). Even though the works related to schema and data changes define models for temporal data warehouses, what is still missing is a conceptual model that allows decision-making

Figure 1. An example of a multidimensional schema for a temporal data warehouse



users to represent data requirement for temporal data warehouses. This model should allow specifying multidimensional elements, i.e., dimensions, hierarchies, facts, and measures, and allow users to clearly indicate which elements they want to be time invariant and for which the changes in time should be kept.

MAIN FOCUS

In this section we present the MultiDim model, a conceptual multidimensional model (Malinowski & Zimányi, 2008a)[2] extended with temporal support (Malinowski & Zimányi, 2006a, 2006b). This model allows users and designers to represent at the conceptual level all elements required in temporal data warehouse applications.

Multidimensional Model for Temporal Data Warehouses

The MultiDim model supports different temporality types: lifespan (LS), valid time (VT), transaction time (TT), and bitemporal time (BT) coming from source systems (if available) and additionally, loading time (LT) generated in a data warehouse.

To describe our model we use the example shown in Figure 1. It includes a set of levels organized into dimensions and a fact relationship. A *level* corresponds to an entity type in the entity-relationship model and represents a set of instances called *members* that have common characteristics. For example, Product, Category, and Store are some of the levels in Figure 1. Levels contain one or several *key attributes* (underlined in Figure 1), identifying uniquely the members of a level, and may also have other *descriptive* attributes.

A *temporal level* is a level for which the application needs to keep the lifespan of its members (e.g., inserting or deleting a product). A *temporal attribute* is an attribute that keeps the changes in its value (e.g., changing a product's size) and the time when they occur. For example, the Product level in Figure 1 is a temporal level that includes temporal attributes Size and Distributor.

A *fact relationship* expresses the focus of analysis and represents an n-ary relationship between levels. For example, the Sales fact relationship between the Product, Store, and Client levels in Figure 1 is used for analyzing sales in different stores.

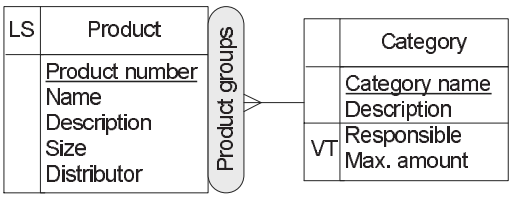
A fact relationship may contain attributes commonly called *measures*. They contain data (usually numerical) that are analyzed using the different dimensions. For example, the Sales fact relationship in Figure 1 includes the measures Quantity and Amount. In the MultiDim model measures are *temporal*, i.e., they always require a temporality type (VT, TT, BT, and/or LT).

A *dimension* allows to group data that shares a common semantic meaning within the domain being modeled, e.g., all data related to a product. It is composed of either a level or one or more hierarchies. For example, Client in Figure 1 is a one-level dimension.

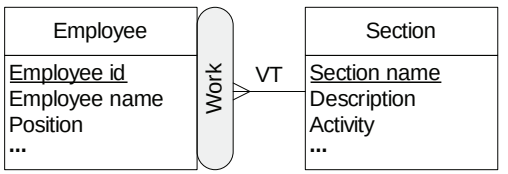
Hierarchies are required for allowing users to analyze data at different levels of detail. A hierarchy contains several related levels, e.g., Product and Category in Figure 1. Given two related levels of a hierarchy, one of them is called *child* and the other *parent* depending on whether they include more detailed or more general data, respectively. In Figure 1, the Product level is a child level while the Category level is a parent level. Key attributes of a parent level define how child members are grouped for the roll-up operation. For example, in Figure 2 since Category name is the key of the Category level, products will be grouped according to the category to which they belong.

The relationships composing the hierarchies are called *child-parent relationships*. These relationships are characterized by *cardinalities* that indicate the minimum and the maximum number of members in one level that can be related to a member in another level. Child-parent relationships may include temporal support. For example, in Figure 1 the LS symbol between Product and Category indicates that the evolution on time of assignments of products to categories will be kept. Temporal support for relationships leads to two interpretations of cardinalities. The *snapshot cardinality* is valid at every time instant whereas the *lifespan cardinality* is valid over the entire member's lifespan. The former cardinality is represented using the symbol indicating temporality type next to the link between levels while the lifespan cardinality includes the LS symbol surrounded by a circle. In Figure 1, the snapshot cardinality between Product and Category levels is many-to-one while the lifespan cardinality is many-to-many. They indicate that a product belongs to only one category at every time instant but belongs to many categories over its lifespan, i.e., its assignment to categories may change. The relationship between levels may include different temporality types: LS, TT, combination of both, and/or LT.

Figure 2. Examples of temporal hierarchies: a) non-temporal relationship between a temporal and a non-temporal levels, and b) temporal relationship between non-temporal levels



(a)



(b)

Since hierarchies in a dimension may express different conceptual structures used for analysis purposes, we use a *criterion name* to differentiate them, such as Product groups or Sales organization in Figure 1.

Modeling Temporal Aspects

Our model supports temporality in an orthogonal way, i.e., hierarchies may contain temporal or non-temporal levels associated with temporal or non-temporal links. Similarly, temporal or non-temporal levels may have temporal or non-temporal attributes. This approach differs from the one used in many temporal models. We consider that it is important to allow users to choose which elements should be temporal or not.

For example, Figure 2 (a) represents a hierarchy composed by a temporal level (Product) and a non-temporal level (Category) with a temporal attribute associated with a non-temporal link. Therefore, the lifespan of products as well as the changes of responsible and maximum amount of categories are kept; on the other hand, other attributes of the levels either do not change or only the last modification is kept. The example in Figure 2 (b) shows a hierarchy with two non-temporal levels associated with a temporal link. This keeps track of the evolution of relationships between employees and sections but we do not store the changes in levels, e.g., when an employee changes its position.

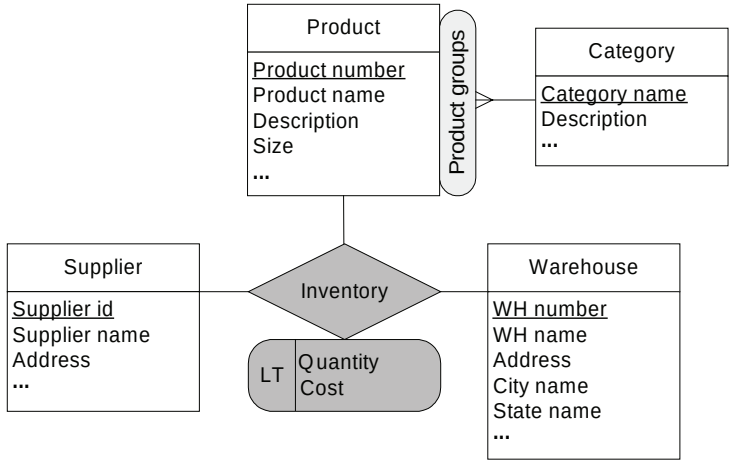
However, to ensure correct management of hierarchies in temporal data warehouses and avoid dangling references, i.e., references to non-existing elements, several constraints should be ensured (Malinowski & Zimányi, 2006a).

Another characteristic of our model is that it uses a consistent approach for providing temporal support for the different elements of a multi-dimensional model, i.e., for levels, hierarchies, and measures. Our model avoids mixing two different approaches where dimensions include explicit temporal support while measures require the presence of the traditional time dimension for keeping track of changes. Since measures are attributes of fact relationships, we provide temporal support for them in the same way as it is done for levels' attributes.

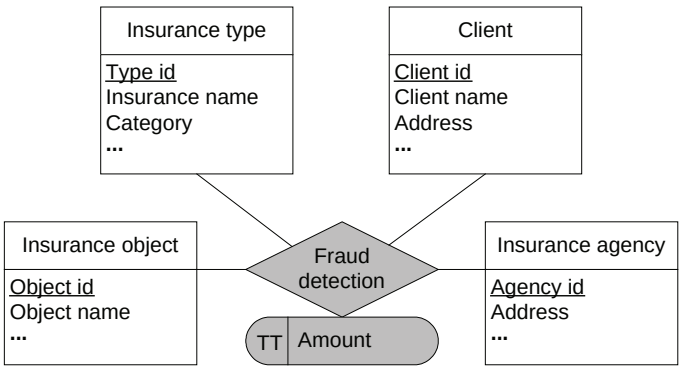
An important question is thus whether it is necessary to have a time dimension in the schema when including temporality types for measures. If all attributes of the time dimension can be obtained by applying time manipulation functions, such as the corresponding week, month, or quarter, this dimension is not required anymore. However, in some temporal data warehouse applications this calculation can be very time-consuming, or the time dimension contains data that cannot be derived, e.g., events such as promotional seasons. Thus, the time dimension is included in a schema depending on users' requirements and the capabilities provided by the underlying DBMS.

Another aspect is the inclusion of different temporality types for measures. The usual practice in temporal data warehouses is to associate valid time support with measures. However, different temporal support can be available in source systems. In Malinowski and Zimányi (2006b), we present several real-world scenarios that include different temporality types for measures enriching the analysis spectrum. The examples in Figure 3 show simplified schemas that include, respectively, loading time and transaction time for measures. The former is used when users require the history of how source data has evolved, but sources are either non-temporal or temporal support is implemented in an ad-hoc manner that can be both inefficient and difficult to automate (Yang & Widom, 1998). The schema in Figure 3 b) is used for an insurance company having as analysis focus the amount of insurance payments. Such a schema

Figure 3. Examples of schemas with a) LT and b) TT support for measures



(a)



(b)

could be used, e.g., to track internal frauds that modify the amount of insurance paid to clients, since the time when the measure values change is kept.

In Malinowski and Zimányi (2006b) we discuss several problems that may occur when data from source systems is aggregated before being loaded into temporal data warehouses. These include different time granularities between source systems and a data warehouse, measure aggregations with different time granularities, and temporal support for aggregated measures.

FUTURE TRENDS

An important issue in temporal data warehouses is the aggregation in the presence of changes in dimension data, schema, or both. Different solutions have already been proposed (e.g., Eder *et al.*, 2002; Mendelzon & Vaisman, 2003; Wrembel & Bebel, 2007; Body *et al.*, 2003). Nevertheless, these proposals require specific software and query languages for implementing and manipulating multidimensional data that vary over time. Further, they do not consider different aspects as mentioned in this paper, e.g., different temporal support in hierarchies and measures.

Another issue is the implementation of temporal data warehouses in current DBMSs, which do not yet provide temporal support. To our knowledge, there are very few proposals for implementing temporal data warehouses in current DBMSs (e.g., Martín & Abelló, 2003; Malinowski & Zimányi, 2006c; Ravat *et al.*, 1999; Mendelzon & Vaisman, 2003). However, only the latter authors provide manipulation features for the proposed structures.

CONCLUSION

Combining the two research areas of data warehouses and temporal databases, allows one to combine the achievements of each of them leading to the emerging field of temporal data warehouses. The latter raises several research issues, such as the inclusion of different temporal support, conceptual modeling, and measure aggregations, among others.

In this paper, we first proposed the inclusion of four different temporality types: three of them come from source systems (if they are available), i.e., valid time, transaction time (or combination of both), and lifespan. A new temporality type, called loading time, is generated in temporal data warehouses. It indicates when data was stored in a temporal data warehouse.

We also presented a conceptual model that is able to express users' requirements for time-varying multidimensional data. The MultiDim model allows temporal support for levels, attributes, relationships between levels forming a hierarchy, and measures. For temporal hierarchies and measures, we discussed different issues that are relevant to ensure the correct data management in temporal data warehouses.

The inclusion of temporality types in a conceptual model allows users, designers, and implementers to include temporal semantics as an integral part of temporal data warehouses. In this way, temporal extensions offer more symmetry to multidimensional models representing in a symmetric manner changes and the time when they occur for all elements of a data warehouse. Since conceptual models are platform independent, logical and physical models can be derived from such a conceptual representation.

REFERENCES

Body, M., Miquel, M., Bédard, Y., & Tchounikine, A. (2003). Handling Evolution in Multidimensional Structures. *Proceedings of the 19th International Conference on Data Engineering*, pp. 581-592. IEEE Computer Society Press.

Bruckner, R. & Tjoa, A. (2002). Capturing Delays and Valid Times in Data Warehouses: Towards Timely Consistent Analyses, *Journal of Intelligent Information Systems*, *19*(2), pp. 169-190.

Eder, J., Koncilia, Ch., & Morzy, T. (2002). The COMET Metamodel for Temporal Data Warehouses. *Proceedings of the 14th International Conference on Advanced Information Systems Engineering*, pp. 83-99. Lecture Notes in Computer Science, N° 2348. Springer.

Golfarelli, M., Lechtenbörger, J., Rizzi, S., & Vossen, V. (2006). Schema Versioning in Data Warehouses: Enabling Cross-Version Querying Via Schema Augmentation. *Data & Knowledge Engineering*, *59*(2), pp. 435-459.

Jensen, C.S. & Snodgrass, R. (2000). Temporally Enhanced Database Design. In M. Papazoglou, S. Spaccapietra, & Z. Tari (Eds.), *Advances in Object-Oriented Data Modeling*, pp. 163-193. Cambridge, MIT Press.

Kimball, R., & Ross, M. (2002). *The Data Warehouse Toolkit.* John Wiley & Sons Publishers.

Malinowski, E. & Zimányi, E. (2008a). Multidimensional Conceptual Models, *in this book*.

Malinowski, E. & Zimányi, E. (2008b). *Advanced Data Warehouse Design: From Conventional to Spatial and Temporal Applications*. Springer.

Malinowski, E. & Zimányi, E. (2006a). A Conceptual Solution for Representing Time in Data Warehouse Dimensions. *Proceedings of the 3rd Asia-Pacific Conference on Conceptual Modelling*, pp. 45-54. Australian Computer Society.

Malinowski, E. & Zimányi, E. (2006b). Inclusion of Time-Varying Measures in Temporal Data Warehouses. *Proceedings of the 8th International Conference on Enterprise Information Systems*, pp. 181-186.

Malinowski, E. & Zimányi, E. (2006c). Object-Relational Representation of a Conceptual Model for Temporal Data Warehouses. *Proceedings of the 18th International Conference on Advanced Information Systems Engineering*, pp. 96-110. Lecture Notes in Computer Science, N° 4001. Springer.

Martín, C. & Abelló, A. (2003). A Temporal Study of Data Sources to Load a Corporate Data Warehouse. *Proceedings of the 5th International Conference on Data Warehousing and Knowledge Discovery*, pp. 109-118. Lecture Notes in Computer Science, N° 2737. Springer.

Mendelzon, A. & Vaisman, A. (2003). Time in Multidimensional Databases. In M. Rafanelli (Ed.), *Multi-*

dimensional Databases: Problems and Solutions, pp. 166-199. Idea Group Publishing.

Moon, B., Vega, F., & Immanuel, V. (2003). Efficient Algorithms for Large-Scale Temporal Aggregation. *IEEE Transactions on Knowledge and Data Engineering*, *15*(3), pp. 744-759.

Ravat, F. & Teste, F. (2006). Supporting Changes in Multidimensional Data Warehouses. *International Review of Computer and Software*, *1*(3), pp. 251-259.

Wrembel, T. & Bebel, B.(2007). Metadata Management in a Multiversion Data Warehouse. In S. Spaccapietra (Ed.), *Journal on Data Semantics, VIII*, pp. 118-157. Lecture Notes in Computer Science, N° 4380. Springer

Yang, J. & Widom, J. (1998). Maintaining Temporal Views Over Non-Temporal Information Source for Data Warehousing. *Proceedings of the 6th International Conference on Extending Database Technology*, pp. 389-403. Lecture Notes in Computer Science, N° 1377. Springer.

KEY TERMS

Bitemporal: A combination of both transaction and valid time.

Conceptual Model: A model for representing schemas that are designed to be as close as possible to users' perception, not taking into account any implementation considerations.

Loading Time: A temporal specification that keeps the information when a data element is stored in a data warehouse.

Granularity: A partitioning of a domain in groups of elements where each group is perceived as an indivisible unit (a granule) at a particular abstraction level.

Lifespan: The record of the evolution of the membership of an instance into its type.

Multidimensional Model: A model for representing the information requirements of analytical applications. A multidimensional model comprises facts, measures, dimensions, and hierarchies.

Temporality Types: Different temporal support that can be provided by a system. They include transaction time, valid time, lifespan, and loading time.

Transaction Time: A temporal specification that keeps the information on when a data element is stored in and deleted from the database.

Valid Time: A temporal specification that keeps information on when a data element stored in the database is considered valid in the perceived reality from the application viewpoint.

ENDNOTES

1 They are usually called time dimensions; however, we use the term "dimension" in the multidimensional context.

2 We only include some elements of the MultiDim model in order to focus on its temporal extension.

Text Categorization

Megan Chenoweth
Innovative Interfaces Inc., USA

Min Song
New Jersey Institute of Technology, USA

INTRODUCTION

Text categorization (TC) is a data mining technique for automatically classifying documents to one or more predefined categories. This paper will introduce the principles of TC, discuss common TC methods and steps, give an overview of the various types of TC systems, and discuss future trends.

TC systems begin with a group of known categories and a set of training documents already assigned to a category, usually by a human expert. Depending on the system, the documents may undergo a process called dimensionality reduction, which reduces the number of words or features that the classifier evaluates during the learning process. The system then analyzes the documents and "learns" which words or features of each document caused it to be classified into a particular category. This is known as supervised learning, because it is based on human knowledge of the categories and their criteria. The learning process results in a classifier which can apply the rules it learned during the training phase to additional documents.

PREPARING THE CLASSIFIER

Before classifiers can begin categorizing new documents, there are several steps required to prepare and train the classifier. First, categories must be established and decisions must be made about how documents are categorized. Second, a training set of documents must be selected. Finally and optionally, the training set often undergoes dimensionality reduction, either through basic techniques such as stemming or, in some cases, more advanced feature selection or extraction. Decisions made at each point in these preparatory steps can significantly affect classifier performance.

TC always begins with a fixed set of categories to which documents must be assigned. This distinguishes TC from text clustering, where categories are built on-the-fly in response to similarities among query results. Often, the categories are a broad range of subjects or topics, like those employed in some of the large text corpora used in TC research such as Reuters news stories, websites, and articles (for example, as the Reuters and Ohsumed corpora are used in Joachims [1998]). Classifiers can be designed to allow as many categories to be assigned to a given document as are deemed relevant, or they may be restricted to the top k most relevant categories. In some instances, TC applications have just one category, and the classifier learns to make yes/no decisions about whether documents should or should not be assigned to that category. This is called binary TC. Examples include authorship verification (Koppel & Schler, 2004) and detecting system misuse (Zhang & Shen, 2005).

The next step in preparing a classifier is to select a training set of documents which will be used to build the classification algorithm. In order to build a classifier, a TC system must be given a set of documents that are already classified into the desired categories. This training set needs to be robust and representative, consisting of a large variety of documents that fully represent the categories to be learned. TC systems also often require a test set, an additional group of pre-classified documents given to the classifier after the training set, used to test classifier performance against a human indexer. Training can occur all at once, in a batch process before the classifier begins categorizing new documents, or training can continue simultaneously with categorization; this is known as online training.

One final, optional step for TC systems is to reduce the number of terms in the index. The technique for doing so, known as dimensionality reduction, is taken from research in information retrieval and is applicable to many data mining tasks. Dimensionality reduction entails reducing the number of terms (i.e., dimensions in the vector space model of IR) in order

to make classifiers perform more efficiently. The goal of dimensionality reduction is to streamline types of classifiers such as naïve Bayes and kNN classifiers, which employ other information retrieval techniques such as document similarity. It can also address the problem of "noisy" training data, where similar terms (for example, forms of the same word with different suffixes) would otherwise be interpreted by the classifier as different words.

Standard techniques for reducing the number of index terms, such as stemming and removing stop words, can address some of these issues. However, more advanced forms of dimensionality reduction are required to more closely simulate the human ability to understand relationships between terms, such as phrases, synonymy, and the importance of particular terms. Berger et al.'s (2006) PARTfs classifier used decision trees, normally a method for building classifiers, as a tool for identifying the phrases that contribute strongly to classification decisions. Another, more advanced, technique is latent semantic indexing (LSI), a natural language processing technique that calculates the synonymy of terms based on their co-occurrence in similar documents in the document corpus. Theoretically, employing LSI creates a classifier that operates on concepts instead of terms; even if terms do not co-occur in a particular document, they can be perceived by the system as referring to the same idea. Cristianini et al. (2002) were the first to develop an SVM classifier that employed LSI. LSI is receiving a great deal of interest in other areas of IR (Dumais, 2004) and shows promise for improving TC in the future.

TYPES OF CLASSIFIERS

Several types of classifiers have been proposed in the literature on text categorization. There are five types of classifier that appear in the majority of the literature and exemplify most current TC scholarship. Those five types are: naïve Bayes, the Rocchio method, *k* nearest neighbor (kNN), decision trees, and support vector machines (SVM).

Naïve Bayesian classifiers (McCallum & Nigam, 1998) calculate the probability that a document belongs to a particular category based on the presence of the same index terms in other documents assigned to that category. For example, if most documents containing the terms "information" and "retrieval" belong to a category C, other documents containing those terms are likely to be categorized in C as well. Classifiers are referred to as "naïve" because they assume that the occurrence of each term is independent of any other term; in other words, they do not account for phrases unless additional feature selection techniques are applied. Overall, naïve Bayesian classifiers perform relatively weakly compared to other methods (Joachims, 1998; Yang & Liu, 1999).

Another common classifier is the Rocchio method, which uses training data to construct a profile of documents that fit a particular category. A Rocchio classifier consists of a list of positive terms and a list of negative terms for each category, and a document is categorized based on the presence and/or absence of these terms (Hull, 1994). For example, the classifier is trained to learn that positive terms for the category "finance" include "bank" and "money," and that "river" is a negative term. New documents containing the terms "bank" and "money" but not "river" will be categorized as "finance." The algorithm for calculating similarity can be adjusted to weigh positive or negative examples more strongly. If the weight for the negative examples is set to zero, the profile of the category can be thought of as the "centroid" of the training documents classified in that category—a cluster of points in the term space where all of the positive examples are positioned. A Rocchio-like classifier proposed by Han and Karypis (2000) was found to outperform a number of other classifiers, including Bayesian, in a single-label categorization task.

kNN classifiers (Yang, 1994) assign a document to one or more categories by finding the *k* most similar documents in the training set. They are sometimes called "lazy learners" because they do not learn the categories and their criteria during the training process, but at the time when a new document is to be categorized. New documents are compared to the other categorized documents within the system. The categories assigned to those documents are taken as candidate categories for the new document, and the similarity scores are the category weights. In an evaluation of five different types of text classifiers, Yang and Liu (1999) found that kNN was one of the top performers in terms of both recall and precision when tested on a corpus of news stories from Reuters, although they also found that performance was significantly improved when multiple categories should be assigned to a document.

Naïve Bayes, Rocchio, and kNN classifiers are all

based on the vector space model from information retrieval, where each term in the document is a factor in classification decisions. As a result, they are particularly susceptible to the dimensionality problem and are often slow and computationally expensive. A more efficient classifier would be able to select and consider only the features that are meaningful to the categorization of a particular document. These classifiers, which include decision tree classifiers and support vector machines, are less susceptible to the problems of high-dimensional term space.

Decision tree classifiers hierarchically organize documents along a tree, each leaf of which represents a category. The structure was proposed in TC by Lin et al (1994), who developed the TV (telescoping vector) tree model. TV trees identify the minimum number of features necessary to classify documents into a given set of categories and make these features nodes on a decision tree. Documents to be classified are compared to each node and divided hierarchically into the classes represented by the ends of each branch (Figure 1). When documents are added that require additional features to discriminate them, more nodes are added to the tree as needed. Because classification uses only the minimum possible number of features required for accuracy, TV trees mitigate the problem of high dimensionality of term space. However, one criticism of decision trees is that they are susceptible to overfitting, or developing classification rules that are specifically suited to the training documents but cannot be generalized to a full document corpus. Training of decision trees often requires an optimization or "pruning" step, in which overly specific branches are removed from the tree (Berger et al, 2006).

Currently, the most popular and commonly used classifiers are support vector machines (SVM) (Joachims, 1998). SVM classifiers find the best decision surface (a hyperplane) in term space between terms that contribute to a document being classified into a particular category and those that do not. Then they use only the terms nearest to that decision surface (called "support vectors") to decide whether a new document belongs to the category. Joachims (1998) compared SVM to four other types of classifier and found that SVM outperformed all four, with kNN being the second best performer. Since Joachims's introduction of SVM to text categorization, many other studies have treated SVMs as the benchmark for performance in the field and have focused on finding ways to enhance SVM. One problem with SVM, however, is that it is particularly susceptible to noisy training data (i.e., errors in the training data, such as those that result from human classification errors). Chen et al. (2005) claimed that this causes SVM not to perform well outside experimental settings and introduced a technique to filter out noise in the training data by identifying and eliminating outliers (positives that are not similar to other positives) before training the SVM. Zhang and Shen (2005) refined SVM to make it more applicable to environments where real-time training and classification are crucial. They used SVM adaptations intended to compensate for noisy training data and further modified the classifier using a technique called online SVM. Online SVM allows the classifier to be trained document by document instead

Figure 1. A simple decision tree

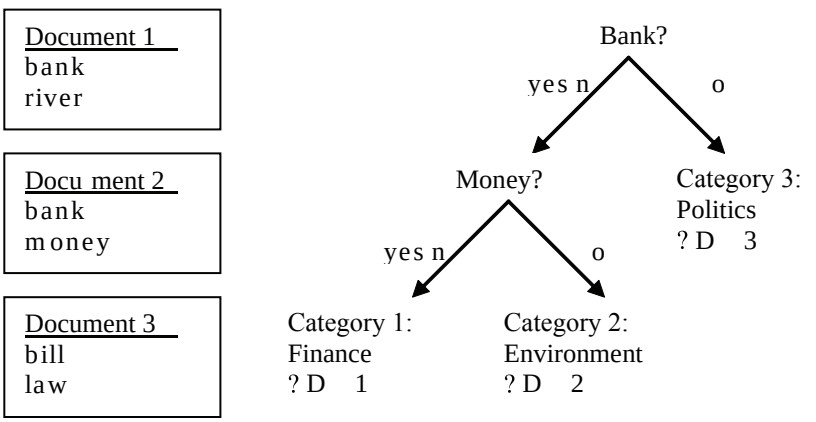


Table 1. Comparison of TC algorithms

Classifier	Method	Performance
Naïve Bayes	Calculates probability of a document belonging to a category based on the presence of the same words in other documents in that category.	Poor compared to other models.
Rocchio Method	Compares document to a list of positive and negative terms for each category and classifies according to presence or weight of those terms.	Poor, especially at classifying into categories with many representative terms.
k Nearest Neighbor	Finds *k* most similar documents and assigns document to the category(ies) those documents are assigned to.	Good, especially with multiple category assignments, but slow, because each document must be compared to all others.
Decision Tree	Separates documents hierarchically in a tree structure, where each node is a relevant term and the end of each branch is a category.	Good, but requires optimization to correct overfitting.
Support Vector Machines	Delineates between terms that do and do not contribute to a document being assigned to a particular category. Categorizes based on presence of the terms that contribute.	Currently the best method, although highly susceptible to errors in training data.

of in a batch, reducing the need for the classifier to be trained offline.

FUTURE TRENDS

With the explosive growth of unstructured data corpora on the Internet, we can expect a growing emphasis on developing new techniques that improve the accuracy and efficiency of TC. Since there is already a proliferation of classifier types that perform relatively well, research in the near future should focus on strategies to improve these techniques. On the practical side, we can also expect to see adaptations of TC methods to familiar systems such as search engines and document indexing systems.

Recent research suggests a trend toward making classifier performance more accurate, efficient, and dynamic through several techniques. As we have seen, high dimensionality is the main cause of many problems for some types of classifiers. Therefore, the most promising strategy for improving performance is to improve techniques for dimensionality reduction. Recent studies using distributional clustering (Al-Mubaid & Umair, 2006), latent semantic analysis (Ishii et al., 2006), and a feature extraction method called angular measures (Combarro et al., 2006) suggest that dimensionality reduction will continue to be a major research focus in the future.

Another significant strategy for improving classifier performance is to reduce the amount of supervised learning required to train a classifier. This procedure is sometimes called bootstrapping. Not only does bootstrapping make training faster, it also opens up more possibilities for the environments in which TC can be applied because it requires fewer manually classified examples from which the classifier can learn. Gliozzo et al. (2005) recently approached this problem by supplying the system with a few terms that exemplify each category instead of supplying a set of manually classified documents. Fung et al. (2006) have attempted to improve classifier performance on unlabeled training examples. Since the vast majority of text corpora contain very few, if any, manually labeled examples, reducing the amount of supervision required is a critical research area for the future of TC.

One final area for future TC research is to develop practical applications that employ TC methods in the types of environments where unstructured text data is currently found. These applications may include Web search engines, digital libraries, and corporate intranets among others. For several years, researchers have been working on search engines that can categorize search results; Chen et al.'s (2002) "personal view agent,"

which can learn which categories a user is interested in, is an early example of such an attempt. More recently, Kules et al. (2006) suggested a "lean" classifier that can use the metadata available in existing digital libraries such as the Open Directory Project to categorize search results. For interest in TC to be sustained, research must continue on practical applications to compete with other commercially available products that employ similar techniques, such as clustering.

CONCLUSION

Text categorization simulates human classification in a wide range of decision tasks, from subject classification to spam filtering. TC begins with a group of categories and a set of pre-classified documents from which a classifier learns to make categorization decisions for additional documents. The major types of classifiers include naïve Bayesian, Rocchio, *k* nearest neighbor, decision trees, and Support Vector Machines; of these, SVM classifiers show the most promise in terms of accurate and efficient performance. With the proliferation of promising TC algorithms, future research should focus on making these algorithms more efficient and accurate, reducing the amount of supervision required in training, and developing practical applications for use on existing corpora of unstructured text data.

REFERENCES

Al-Mubaid, H., & Umair, S. (2006). A new text categorization technique using distributional clustering and learning logic. *IEEE Transactions on Knowledge and Data Engineering 18*(9), 1156-1165.

Berger, H., Merkl, D., & Dittenbach, M. (2006). Exploiting partial decision trees for feature subset selection in email categorization. In *Proceedings of the 21st Annual ACM Symposium on Applied Computing* (Dijon, France), 1105-1109.

Chen, C.C., Chen, M.C., & Sun, Y.L. (2002). PVA: A Self-Adaptive Personal View Agent. *Journal of Intelligent Information Systems 18*(2/3): pp. 173-194.

Chen, L., Huang, J., & Gong, Z.H. (2005). An anti-noise text categorization method based on support vector machines. In *Advances in Web Intelligence: Proceedings of the Third International Atlantic Web Intelligence Conference* (Lodz, Poland, 2005), pp. 272-278.

Combarro, E.F., Montañés, & Ranilla, J. (2006). Angular measures for feature selection in text categorization. In *Proceedings of the 21st Annual ACM Symposium on Applied Computing* (Dijon, France), 826-830.

Cristianini, N., Shawe-Taylor, J., & Lodha, H. (2002). Latent semantic kernels. *Journal of Intelligent Information Systems 18*(2/3), pp. 127-152.

Dumais, S. (2004). Latent semantic analysis. *Annual Review of Information Science and Technology 38*, 188-230.

Fung, G.P.C., Yu, J.X., Lu, H.J., Yu, P.S. (2006). Text classification without negative examples revisit. *IEEE Transactions on Knowledge and Data Engineering 18*(1), 6-20.

Gliozzo, C., Strapparava, A., & Dagan, I. (2005). Investigating unsupervised learning for text categorization bootstrapping. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing* (Vancouver, B.C., 2005), pp. 129-136.

Han, E.H., and Karypis, G. (2000). Centroid-based document classification: analysis and experimental results. In *Proceedings of Data Mining and Knowledge Discovery: Lecture Notes in Artificial Intelligence 1910* (Berlin; Heidelberg: Springer-Verlag), 424-431.

Hull, M. (1994). Improving text retrieval for the routing problem using latent semantic indexing. In *Proceedings of SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval* (Dublin, Ireland), 282–289.

Ishii, N., Murai, T., Yamada, T., & Bao, Y. (2006). Text classification by combining grouping, LSA and kNN. *Proceedings of the 5th IEEE/ACIS International Conference on Computer and Information Science and 1st IEEE/ACIS International Workshop on Component-Based Software Engineering, Software Architecture and Reuse* (Honolulu, Hawaii), 148-154.

Joachims, T. (1998). Text categorization with support vector machines: learning with many relevant features. In *Proceedings of ECML-98, 10th European Conference on Machine Learning* (Chemnitz, Germany, 1998), 137-142.

Koppel, M., & Schler, J. (2004). Authorship verification as a one-class classification problem. In *Proceedings of the 21st International Conference on Machine Learning* (Banff, Canada), 62-68.

Kules, B., Kustanowitz, J., & Shneiderman, B. (2006). Categorizing Web search results into meaningful and stable categories using fast-feature techniques. In *Proceedings of JDCL'06* (Chapel Hill, NC), 210-219.

Lin, K.I., Jagadish, H.V., & Faloutsos, C. (1994). The TV-tree: an index structure for high-dimensional data. *VLDB Journal 3*, 517-542.

McCallum, A., & Nigam, K. (1998). A comparison of event models for naive Bayes text classification. In *Learning for Text Categorization: Papers from the AAAI Workshop* (Madison, Wisconsin). Retrieved March 1, 2006, from http://www.kamalnigam.com.

Sebastiani, F. (2002). Machine learning in automatic text categorization. *ACM Computing Surveys, 34*(1), 1-47.

Yang, Y.M. (1994). Expert network: effective and efficient learning from human decisions in text categorisation and retrieval. In *Proceedings of SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval* (Dublin, Ireland), 13-22.

Yang, Y., & Liu, X. (1999). A re-examination of text categorization methods. *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Berkeley, CA), 42-49.

Zhang, Z.H., & Shen, H. (2005). Application of online-training SVMs for real-time intrusion detection with different considerations. *Computer Communications 28*, 1428-1442.

KEY TERMS

Bootstrapping: Method for reducing the amount of training data required for a TC system by providing the classifier with a few examples from which to learn, instead of a full set of categorized documents.

Dimensionality Reduction: Data mining technique that reduces the number of features (words) in the term space, either through simple techniques such as stemming or by grouping words into phrases or concepts.

Latent Semantic Indexing: Dimensionality reduction technique that reduces dimensions to concepts rather than words by calculating the synonymy of terms based on their co-occurrence in similar documents in the document corpus.

Overfitting: Development of a classifier that is too specific to the training data; fixed by optimization or "pruning."

Text Clustering: Unsupervised learning technique in which a set of documents resulting from a query is categorized on the fly, based on internal similarities, as opposed to by a predefined set of categories as in text categorization.

Vector Space Model: Model for document representation in information retrieval, in which each term is considered a vector or weight in multidimensional term "space."

Text Mining by Pseudo-Natural Language Understanding

Ruqian Lu
Chinese Academy of Sciences, China

INTRODUCTION

Text mining by pseudo natural language understanding (TM by PNLU for short) is a technique developed by the AST group of Chinese Academy of Sciences, as part of the project automatic knowledge acquisition by PNLU, which introduces a partial parse technique to avoid the difficulty of full NLU. It consists of three parts: PNL design, PNL parser implementation and PNLU based automatic knowledge acquisition. Its essence is twofold: a trade-off between information gain and feasibility of parsing, and a rational work division between human and computer,

BACKGROUND

Experts in knowledge engineering have been agreeing on the basic point of view that the most challenging problem in the construction of knowledge based systems is how to acquire enough and high quality domain knowledge. People usually use natural language understanding techniques to acquire knowledge from technical literature. As it was shown by practice, natural language understanding has always been a very difficult problem.

Experiences have shown that many practical applications do not need a complete and perfect understanding of the natural language texts. By lowering down the requirement of NLP a bit and providing the computer with human help a bit we can let the computer process and understand "natural language" texts, including acquiring knowledge from it, massively, automatically and efficiently. This was the goal of developing the technique of TM by PNLU.

MAIN FOCUS

Definition of PNL

Let's use the notation PNL for pseudo natural language and PNLU for PNL understanding. The former denotes a class of languages, while the latter denotes a kind of technique for processing PNL. Generally speaking, PNL looks very similar to natural language, but can be understood, analyzed and compiled by computer to an extent by which it can meet the need of some application, for example compiling a text book in a knowledge base for expert consultation.

Design of a PNL

The process of designing a PNL is as follows:

1. Determine a set of semantic constructs of the application domain. For example, if the domain is mathematics, then the semantic constructs are case frames providing sentence semantics frequently used in mathematics textbooks, like concept definition, theorem proving, exercise presentation, etc.
2. Select a natural language, e.g. English, as background language;
3. Look for sentence patterns in this language, whose meaning corresponds to the semantic constructs selected in the first step. These sentence patterns may look like: **if * then * is called ***; **since * is true and * is not true we can infer from * that * is true**.
4. Organize the set of selected sentence patterns in grammar form and call it the key structure of the language, which is now called pseudo-natural. That means: not every combination of sentence patterns is a legal key structure. All parts marked with stars will be skipped by any PNL parser. We call these parts "don't care".

PNL Grammar

As an example, the following tiny grammar implies the key structure of a general classification statement:

⟨Classification Sentence⟩::=[⟨Leading word⟩<***Don't care***>,]⟨classification leading sentence⟩
⟨Leading word⟩::=**According to | Based on**
⟨classification leading sentence⟩::=⟨***Don't care***⟩⟨classification word⟩[⟨number⟩[**main**]⟨type word⟩[,⟨sequence of ***Don't care***⟩]] | There⟨be word⟩⟨number⟩[**main**]⟨type word⟩of⟨***Don't care***⟩. They are⟨sequence of ***Don't care***⟩.
⟨classification word⟩::=⟨be word⟩**classified into** | ⟨mood word⟩**be classified into**
⟨be word⟩⟩::= **is | are**
⟨type word⟩::= **classes | types | sorts | kinds** ……

This grammar may recognize sentences like: *Blood cells* **are classified into two types,** *red blood cells*, *white blood cells***.**

Parsing a PNL Text

While parsing a PNL text, the computer tries to understand the text, based (and only based) on the semantics of the underlying key structure. This understanding is necessary superficial in the sense that no information other than that implied by the key structure will be gained by the computer. It abstracts away all unnecessary details regarding the current application and thus makes the language understanding much easier. For example: consider the sentence:

If the color of the blood cell is red **than** the blood cell **is called** erythrocyte.

This is a (shallow) definition of red blood cell. But here we can already see the abstraction principle of PNL. With this knowledge, a computer can answer questions like "what is erythrocyte?," "How do we call a blood cell when the color of the blood cell is red?" etc. even without knowing the meaning of "red" or "cell."

Mechanism of TM by PNLU

The mechanism of TM by PNLU can be roughly described as follows:

1. Design a PNL;
2. Implement a compiler, which can parse PNL texts, acquiring knowledge from it and organizing it in a domain knowledge base;
3. Each time when knowledge is to be acquired, use an OCR device to scan the documents into the computer. Modify the scanned texts slightly to turn them in their PNL form;
4. Let the computer parse and analyze the PNL texts and produce a knowledge base (, which may need to be integrated with an existing knowledge base).

A series of PNL in different domains have been developed for automatic knowledge acquisition and system prototyping, including BKDL for expert systems, EBKDL and SELD for ICAI systems, DODL/ BIDL for Management Information Systems and KUML/WKPL for knowware engineering.

Layers of PNL

For the same application domain, one can divide the key structure of a PNL in several layers. There are three basic layers: the core layer, which contains sentence patterns used in all domains; the domain layer, which contains technical expressions used in a particular domain; and the jargon layer, which contains professional expressions used by a particular group of users. Each time when one designs a new PNL, the core layer, which occupies the major part of the key structure, does not have to be modified. Only part of the domain layer should be renewed. The jargon layer is usually very small and does not play an important role. Of course it is also possible to define intermediate layers between the basic layers.

Spectrum of PNL

It is easy to see that if we enlarge the key structure of PNL, then the computer will acquire more detailed knowledge from a PNL text. For example, if we add the sentence pattern "**color of * is ***" to the key structure, then the computer may additionally know that color is an attribute, which can be used to describe physical objects. On the other hand, if we reduce the key structure, then the knowledge acquired by the computer will have a larger granule. In this way, PNL defined with different key structure form a spectrum, which is a partial order. The upper limit of this spectrum is the natural

language, while the lower limit is the formal language consisting of meaningless strings of symbols.

Degree of Pseudo-Naturality

It is interesting to consider the quantitative characterization of this PNL spectrum. We assign to each PNL a degree of pseudo naturality (DPN for short). In this way the PNL spectrum is reduced from a partial order to a total order. There are different ways of defining this degree. One possibility is to define the DPN of each PNL text as the ratio of the number of keywords contained in it to the number of all words in this text. The DPN of the PNL is then the statistically averaged value of DPN of texts written in this PNL. Since it is impossible to count all such texts, an empirical calculation based on a limited set of selected representative texts can be performed to obtain an estimate. Another possibility of defining DPN is to calculate the ratio of cumulative length of keywords to the length of the text itself instead of calculating the ratio of word numbers. According to this definition, the DPN of a natural language is one, whereas the DPN of a formal language is zero. The larger the DPN of a PNL is, the more difficult is its parsing, and the more information we will obtain from parsing its sentences. We estimate the DPN of BKDL between 10% and 15%.

FUTURE TRENDS

In the 2003 special issue of JACM, when looking into the future of computer science research, at least three authors proposed the topic of letting computer read massive texts and acquire knowledge from them. It is to expect that the TM by PNLU technique will be helpful in completing this job.

Future research includes applying PNLU to NL directly; defining rigorous semantics for PNL and a through study on PNL spectrum.

CONCLUSION

A comparison of PNLU vs. NLU technique is given in Table 1.

The first use of TM by PNLU technique is in the work about CONBES and BKDL (1990-1991). The first PNL was BKDL (Book Knowledge Description Language) published in (Lu, R., Cao, C., 1990), which 'incorporates a built-in natural language frame, which makes its programs look very much like Chinese texts while keeping the semantic unambiguity'. In (Lu, R., 1994), another PNL, called BIDL and used in PROMIS system, was defined as "natural-like language". The first appearance of the term "pseudo-natural language" was in (Lu, R., 1994.6). There is only little literature mentioning the term PNL. Among them the two recent examples are Metalog (2004) and ORM-ML (2002). From which the former is used to enrich a markup language, while the latter is used in some query logic. Thus their techniques and results achieved are different from the first 10 references cited after the end of this chapter.

TM by PNLU is a promising technique. However, improvement of this technique is needed for many potential new applications. Web text mining is one of the examples.

Table 1. PNL/PNLU vs. NL/NLU

	Language Generating Structure	This Structure covers	Parsing Procedure works on	Knowledge acquired by TM	Granule of acquired Knowledge	Relation to each other
NL/NLU Technique	A NL grammar	All Sentences of the NL	Every Sentence of a text	Data, Term, Concept, Relation	Usually small	NL is limit of PNL Spectrum
PNL/PNLU Technique	A PNL Key Structure	A subset of the NL	Only key Structure of the text	Theory, Domain Knowledge	Usually large	Each PNL is an approximation of NL

REFERENCES

Aurora, P. P., Rafael, B. L., José, R. S. (2006). Topic Discovery Based on Text Mining Techniques, *Information Processing and Management*, 43(3), 752–768.

Cao, C., Lu, R. (1991). The Knowledge Processing of CONBES, *Advances in Chinese Computer Science*, (III) 25-40, World Scientific, Singapore.

Cao, C., Lu, R. (1991). CONBES, A New Generation Tool for Developing Expert Systems, *Journal of Computers*, 893-901.

Chen,Y. (2000). A Knowledge Acquisition System Based on Pseudo-Natural language Understanding, *Acta of Hua Qiao Univ*, (2).

Feigenbaum, E.A., McCorduck, P. (1983). The Fifth Generation: Artificial Intelligence and Japan's Challenge to the World, Addison-Wesley.

Feigenbaum, E. A.(2003). Some Challenges and Grand Challenges for Computational Intelligence, *JACM*, 50 (1), 32-40.

Feldman, R., Sanger, J. (2007). The Text Mining Handbook-Advanced Approaches in Analyzing Unstructured Data, USA: New York.

Gray, J. (2003). What Next? A Dozen Information-Technology Research Goals, *JACM*, 50 (1), 41-57.

Lu, R., Cao, C. (1990). Towards Knowledge Acquisition from Domain Books, *Current Trends in Knowledge Acquisition*, 289-301, IOC, Amsterdam.

Lu, R. (1994.6). Automatic Knowledge Acquisition by Understanding Pseudo- Natural Languages, Theory and Praxis of Machine Learning, *Dagstuhl Seminar Report 91* (9426), 11-12.

Lu, R., Jin, Z., Wan, R.(1994). A knowledge- based approach for automatically prototyping management information systems, *AVIGNON 94*'.

Lu, R., Cao, C., Chen, Y., Mao, W., Chen, W., Han, Z. (1995). The PLNU approach to automatic generation of ICAI systems, *Science in China*, *series A*, 38 (supplement), 1-11.

Lu, R., Jin, Z., Wan, R. (1995) Requirement specification in pseudo-natural language in PROMIS, *proc. of 19th COMPSAC*, 96-101.

Lu, R., Jin, Z. (2000). Domain Modeling Based Software Engineering: A Formal Approach, Kluwer Publishing Co.

Lu, R.(2007). Knowware, the third Star after Hardware and Software, Polimetrica Publishing Co., Italy.

Marchiori, M.(2004). Towards a People's Web: Metalog, http://www.w3.org/People/Massimo/papers/2004/wi2004.pdf.

OASIS. (2002). STARLab ORM Markup Language (ORM-ML), http://xml.coverpages.org/ORM-200206.pdf.

Reddy, R. (2003). Three Open Problems in AI, *JACM*, 50 (1), 83-86.

Weiss, S. M., Indurkhya N., Zhang T., Damerau F. J.(2005). Text Mining-Predictive Methods for Analyzing Unstructured Information, New York.

KEY TERMS

Degree of Pseudo Naturality (DPN): A measure to reduce the partial order of PNL spectrum to a linear order that may be based on either of the two formulae:

$$DPN1 = \frac{Number(Key-Words-\dot{n}-Text)}{Number(Words-\dot{n}-Text)}$$

$$DPN2 = \frac{Length(Key-Words-\dot{n}-Text)}{Length(Text)}$$

Layers of PNL: Set of subsumpting PNL ordered according to their generality of application. It is often possible to define semantics of higher layer PNL key structure by using that of lower layers.

Key Structure: The grammatical form of legal keyword combination in a PNL.

Pseudo Natural Language (PNL): Subset of a natural language with a predefined key structure.

Pseudo Natural Language Understanding (PNLU): Parsing of a PNL by only analyzing the syntax, semantics and pragmatics of key structure contained in it.

Spectrum of PNL: Partial ordering of all PNL according to subsumption of their key structures.

Text Mining by PNLU (TM by PNLU): Using PNLU techniques to acquire knowledge from written documents automatically.

Text Mining for Business Intelligence

Konstantinos Markellos
University of Patras, Greece

Penelope Markellou
University of Patras, Greece

Giorgos Mayritsakis
University of Patras, Greece

Spiros Sirmakessis
Technological Educational Institution of Messolongi and Research Academic Computer Technology Institute, Greece

Athanasios Tsakalidis
University of Patras, Greece

INTRODUCTION

Nowadays, business executives understand that timely and accurate knowledge has become crucial factor for making better and faster business decisions and providing in this way companies a competitive advantage. Especially, with the vast majority of corporate information stored as text in various databases, the need to efficiently extract actionable knowledge from these assets is growing rapidly. Existing approaches are incapable of handling the constantly increasing volumes of textual data and only a small percentage can be effectively analyzed.

Business Intelligence (BI) provides a broad set of techniques, tools and technologies that facilitate management of business knowledge, performance, and strategy through automated analytics or human-computer interaction. It unlocks the "hidden" knowledge of the data and enables companies to gain insight into better customers, markets, and business information by combing through vast quantities of data quickly, thoroughly and with sharp analytical precision.

A critical component that impacts business performance relates to the evaluation of competition. Measurement and assessment of technological and scientific innovation and the production of relative indicators can provide a clear view about progress. Information related to those activities is usually stored to large databases and can be distinguished in: research information stored in publications or scientific magazines and development-production information stored in patents.

Patents are closely related to *Technology Watch*, the activity of surveying the development of new technologies, of new products, of tendencies of technology as well as measuring their impact on actual technologies, organizations or people. Statistical exploitation of patent data may lead to useful conclusions about technological development, trends or innovation (Chappelier et al., 2002).

Traditional methods of extracting knowledge from patent databases are based on manual analysis carried out by experts. Nowadays, these methods are impractical as patent databases grow exponentially. *Text Mining (TM)* therefore corresponds to the extension of the more traditional Data Mining approach to unstructured textual data and is primarily concerned with the extraction of information implicitly contained in collections of documents. The use of automatic analysis techniques allows us to valorize in a more efficient way the potential wealth of information that the textual databases represent (Hotho et al., 2005).

This article describes a methodological approach and an implemented system that combines efficient TM techniques and tools. The BI platform enables users to access, query, analyze, and report the patents. Moreover, future trends and challenges are illustrated and some new research that we are pursuing to enhance the approach are discussed.

BACKGROUND

Patents are closely related to technological and scientific activities (Narin, 1995). They give an indication of the structure and evolution of innovative activities in countries, regions or industries. In this framework, patents are linked to Research and Development (R&D) and can be considered as indicators of R&D activities (Schmoch et al. 1998).

A patent is a legal title granting its holder the exclusive right to make use of an invention for a limited area and time by stopping others from, amongst other things, making, using or selling it without authorization (EPO, 2006). The patent applicant has to provide a detailed technical description of its invention but also mention the points that render it an original application with innovative elements.

A patent can be decomposed and described by several fields (table 1). Each field contains specific information while each patent is described by a code (or in many cases more than one codes) depicting its technical characteristics. These codes are given to patents based on the International Patents Classification system (IPC) or other classification systems. We should also mention that patent documents can be either retrieved from on-line patent databases, or patent databases available on CD-ROMs.

Tools from various vendors provide the user with a query and analysis front-end to the patent data. Some of these tools perform only simple analysis and produce tables, charts or reports e.g. PatentLab II (http://www.wisdomain.com/download.htm), BizInt Smart Charts for Patents 3.0 (http://www.bizcharts.com/patents/index.html), MapOut Pro (http://www.mapout.se/ MapOut.html), etc. Other tools demonstrate enhanced capabilities by using advanced TM techniques e.g. Management and Analysis of Patent Information Text or MAPIT (http://www.mnis.com/mpt.html), VantagePoint (http://www.thevantagepoint.com), Aureka (http://www.micropat.com/static/aureka.htm), Technology Opportunities Analysis or TOA (http://www.tpac.gatech.edu/toa.php), etc.

Table 1. Patent fields in the ESPACE ACCESS database

PN	Priority Number (number of the patent).
AN	Application Number.
PR	Priority Year.
DS	Designated States.
MC	Main Classification Codes.
IC	All Classification.
ET	English Title.
FT	French Title.
IN	Inventor.
PA	Applicant (name of the company depositor).
AB	English Abstract.
AF	French Abstract.

A BUSINESS INTELLIGENCE PLATFORM FOR PATENT MINING

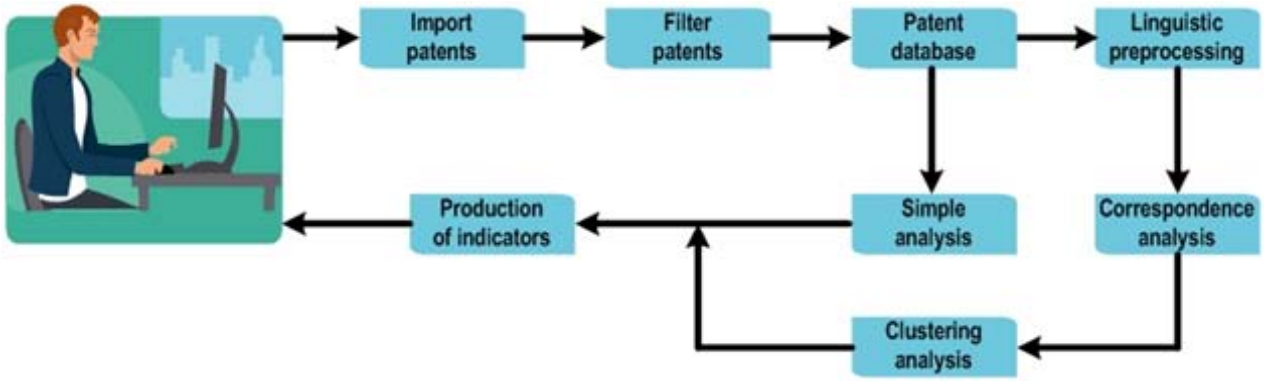
Research and development investment in knowledge discovery and management technologies has made significant progress. However, there still exists a need for an approach that combines efficient and innovative tools for the analysis of patent data, which will guide users (e.g. R&D planners, business analysts, patents analysts, national and international patent offices, economic organizations, national statistical offices, venture capitalists, industrial bodies, etc.) to extract only necessary information and exploit it in an informative way in order to draw useful conclusions.

Our platform was designed to fill this need by quickly analyzing large collections of patents, utilizing multiple algorithms and visualizations, and producing indicators concerning the scientific and technological progress (Markellos et al., 2003). These indicators provide a global understanding of the patent collection and help users to make conclusions about on-going changes and their effects. The methodological approach is depicted in figure 1.

Patents Preparation

The system enables data importing through an easy-to-use dialog box. After downloading a data file in .txt format from MIMOSA search engine a new project to work with can be created. In this step, to reduce the patents representation for efficiency of computation and scalability purposes, while maintaining the maximum of information, several techniques are used. We browse the patent records, read their contents, modify

Figure 1. Methodological approach.



their fields or filter the data in order to prepare the appropriate patents set.

Moreover, we add, delete or modify the patents fields in order to correct the data (by removing html tags or punctuation characters), change the way they appear or eliminate data inconsistencies. For example, a frequent data inconsistency appears in the Assignee (Inventor or Applicant) field where the same company is recorded with one or more different names (i.e. "ABC Co., Ltd" and "ABC Company Limited" is regarded as two different assignees, although they are identical). We should correct any data inconsistencies prior to the analysis task in order to get accurate results. The results of this phase are automatically saved to an internal representation form suitable for further analysis. Moreover, we are able to export the system database in other well known formats e.g. .xls, .doc, .pdf, etc.

User-defined Linguistic Process

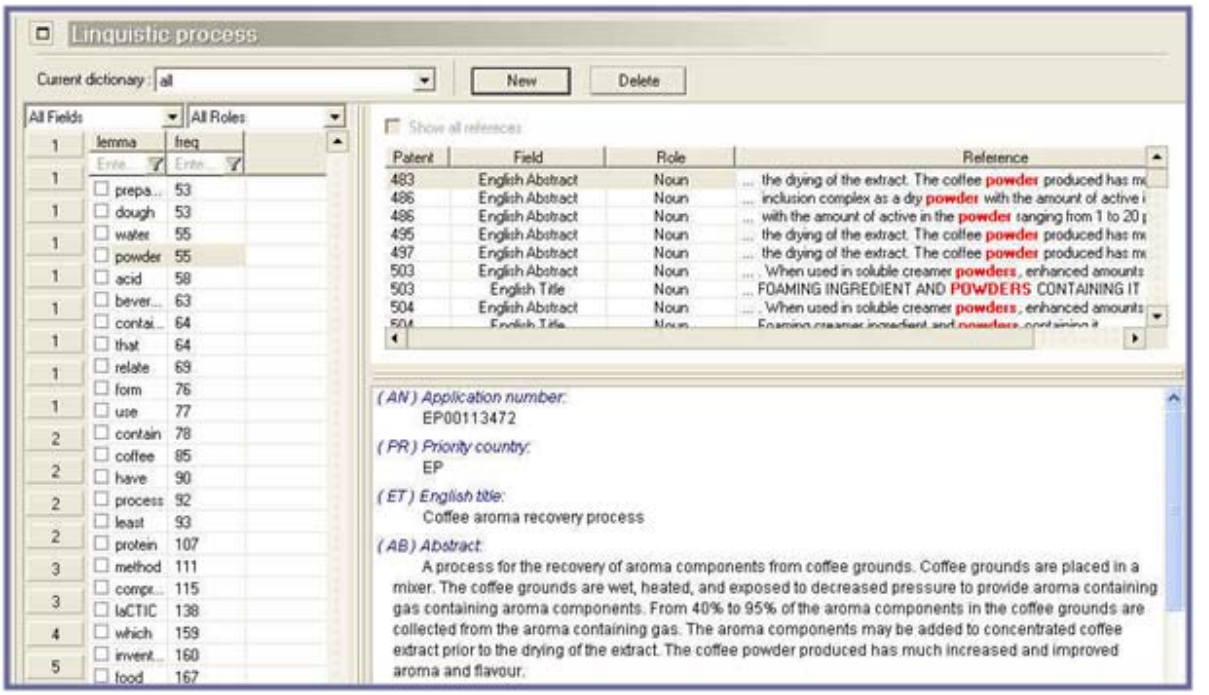
The main aim of the linguistic process (Beaugrande & Dressler, 1981) is to identify the words from specific fields (abstract and title) of each patent, filter out insignificant words (i.e. words like "this", "the") and determine a lexicon of lemmas to support the latter analysis. A parser is used for reading the textual data and restricting the morphologic variation of each word to its unique canonical representation-lemma. We further reduce the vocabulary size, by selecting the word categories, as identified by the assigned parts-of-speech, and restrict the analysis to specific word categories. We can use a number of different dictionaries in the process. Either we select a predefined dictionary or define a new one. We determine the parse fields (title, abstract or both) and specify the word categories to be kept (nouns, verbs, symbols or any combination of them).

In addition, we use stop-lists to eliminate words bearing no content and exclude them from the analysis, such as frequent used words. We utilize synonym lists to collapse semantically similar words and stem variants to their base form. We identify semantically important terms that must remain and involve in the analysis. Moreover, we explore the list of lemmas and the corresponding frequencies by referring back to the original patent data. We can see the list of references for every lemma. A list is automatically created with the number of the patents where the lemma appears, the exact field (English Abstract or English Title), the syntactic category of the lemma (Noun, Verb, etc.) and the reference sentence where the lemma is marked in red color. We export the list of lemmas and the corresponding frequencies in Excel format. Finally, we visualize all the existing words both in order of frequency or in alphabetical order.

Simple Analysis

Simple Analysis based on the original data and more specifically in the supplementary variables involved in the analysis. Therefore, it is applied in a data set that does not need linguistic pre-processing. This procedure gives us the ability to use simple statistical measures for quantifying information stored in patents and representing it in various graphs and tables.

Figure 2. Linguistic processing.



Correspondence Analysis

Correspondence Analysis (Benzecri, 1992) is the first step in order to perform Cluster Analysis on patent data. Previously we should perform linguistic pre-processing. The input data is the contingency table where the rows contain the lemmas, the columns represent the patents, and each cell gives the frequency of the specific word in the corresponding patent. This analysis enables us to explore the non-random dependencies between the variables involved in it and the vocabulary obtained from the title and abstracts of the patents. More precisely, the correspondence analysis produces a new vector space in which similarities between the rows and the columns of the input contingency table (as measured by the x^2-distance) can be visualized as geometric proximities.

- **Words selection.** We can create the contingency table by applying specific criteria and preferences. From the lemmas list, the preferred frequencies (i.e. maximum or minimum) can be selected. The following statistic metric is also available: the frequency of the selected lemmas in normal or logarithmic scale. Moreover, we can change the graphical appearance of the metric.
- **Contingency table.** We can explore the contingency table of the lemmas (rows) involved in the analysis and the corresponding patents (columns). Non-zero frequencies are highlighted, so we can easily focus in those cells.
- **Number of dimensions.** We can define the number of dimensions to be kept as a result of the *Factor Analysis* performed, through an interactive process. In each row the following information is available: 1) the Eigen value for each dimension, 2) the percentage of information, 3) the cumulative percentage of information expressed for the selected dimensions. The cumulative percentage of the information is also graphically visualized through horizontal bars.

Cluster Analysis

Cluster Analysis (Huang, 2006) is the basis for technology indicators derivation and the classification of the technology on-goings in homogeneous classes. It follows linguistic pre-processing of patents and Correspondence Analysis applied to the lemmatized

Figure 3. Words selection procedure

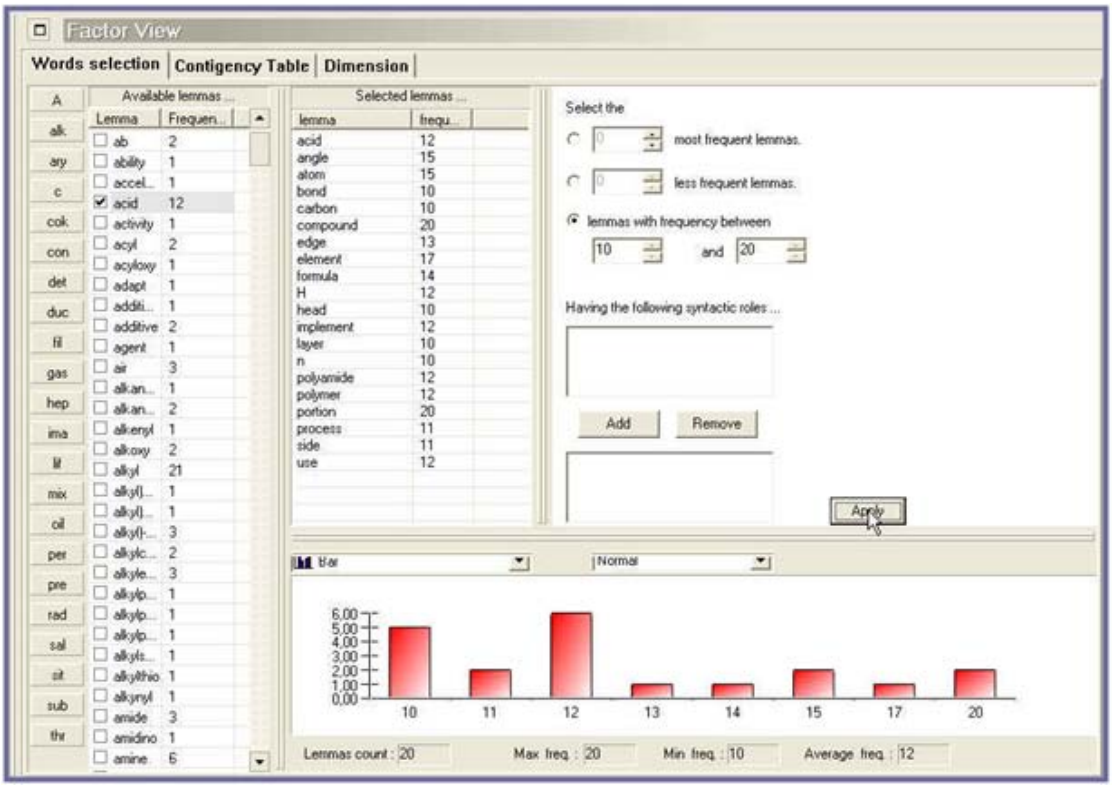
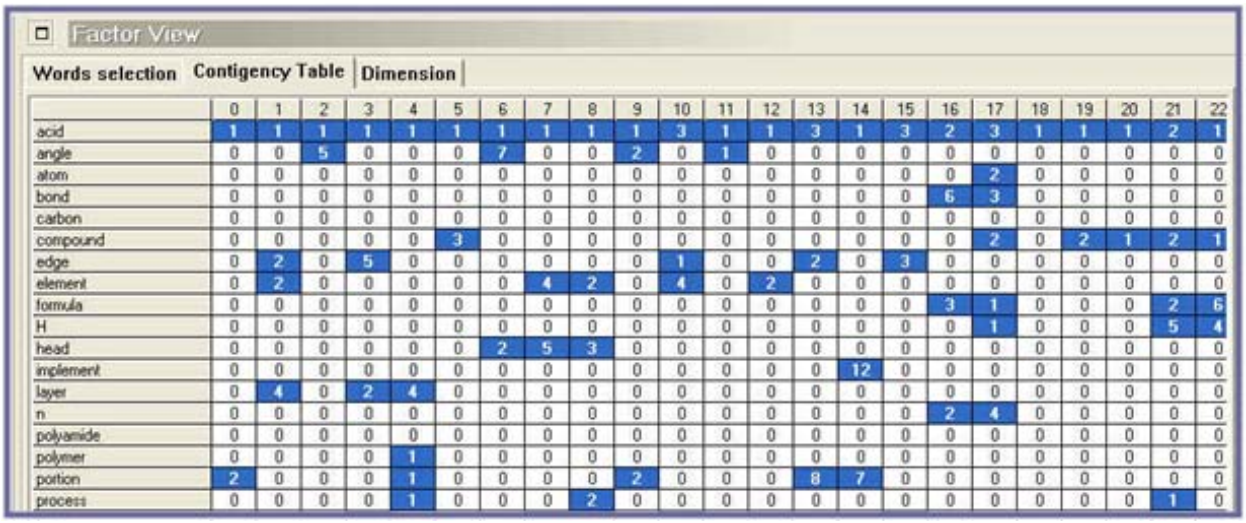


Figure 4. Contingency table

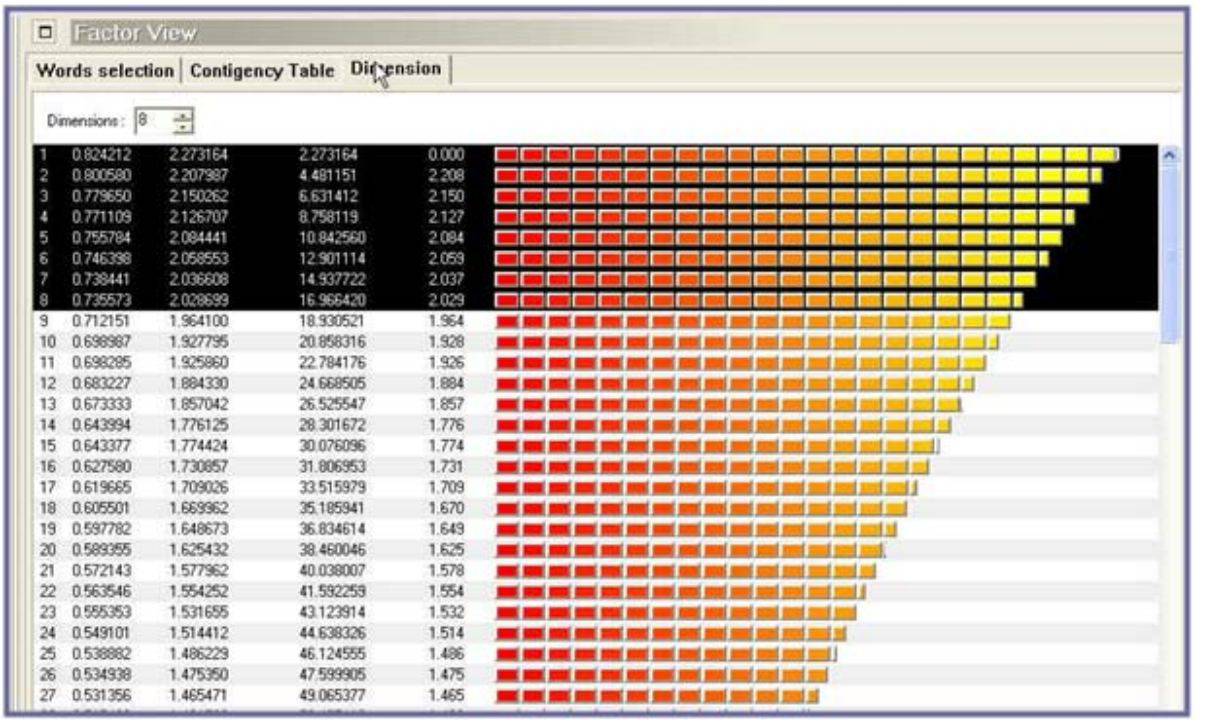


	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
acid	1	1	1	1	1	1	1	1	1	1	3	1	1	3	1	3	2	3	1	1	1	2	1
angle	0	0	5	0	0	0	7	0	0	2	0	1	0	0	0	0	0	0	0	0	0	0	0
atom	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0
bond	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	3	0	0	0	0	0
carbon	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
compound	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	2	0	2	1	2	1
edge	0	2	0	5	0	0	0	0	0	0	1	0	0	2	0	3	0	0	0	0	0	0	0
element	0	2	0	0	0	0	0	4	2	0	4	0	2	0	0	0	0	0	0	0	0	0	0
formula	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	1	0	0	0	2	6
H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	5	4
head	0	0	0	0	0	0	2	5	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
implement	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	0	0	0	0	0	0	0	0
layer	0	4	0	2	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
n	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	4	0	0	0	0	0
polyamide	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
polymer	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
portion	2	0	0	0	1	0	0	0	0	2	0	0	0	8	7	0	0	0	0	0	0	0	0
process	0	0	0	0	1	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	1	0

data. The aim is to identify groups of patents that share common vocabulary and technologies in order to produce conclusions about technological trends and innovation.

The selection of the clustering algorithm based on the number of data and can be either “Hierarchical Clustering” or “k-means Clustering”. In the second case, “Hierarchical Clustering” is performed in the k clusters obtained from the “k-means”. In both cases Cluster Analysis is performed in the factorial axes derived from the Correspondence Analysis. Through the Cluster Analysis the patents are grouped into clus-

Figure 5. Selection of the number of dimensions to be kept



1	0.824212	2.273164	2.273164	0.000
2	0.800580	2.207987	4.481151	2.208
3	0.779650	2.150262	6.631412	2.150
4	0.771109	2.126707	8.758119	2.127
5	0.755784	2.084441	10.842560	2.084
6	0.746398	2.058553	12.901114	2.059
7	0.738441	2.036608	14.937722	2.037
8	0.735573	2.028699	16.966420	2.029
9	0.712151	1.964100	18.930521	1.964
10	0.698987	1.927795	20.858316	1.928
11	0.698285	1.925860	22.784176	1.926
12	0.683227	1.884330	24.668505	1.884
13	0.673333	1.857042	26.525547	1.857
14	0.643994	1.776125	28.301672	1.776
15	0.643377	1.774424	30.076096	1.774
16	0.627580	1.730857	31.806953	1.731
17	0.619665	1.709026	33.515979	1.709
18	0.605501	1.669962	35.185941	1.670
19	0.597782	1.648673	36.834614	1.649
20	0.589355	1.625432	38.460046	1.625
21	0.572143	1.577962	40.038007	1.578
22	0.563546	1.554252	41.592259	1.554
23	0.555353	1.531655	43.123914	1.532
24	0.549101	1.514412	44.638326	1.514
25	0.538882	1.486229	46.124555	1.486
26	0.534938	1.475350	47.599905	1.475
27	0.531356	1.465471	49.065377	1.465

ters that share similar characteristics and thus describe similar technologies.

- **Number of clusters – cut-off value.** From the produced dendrogram, by moving the horizontal weighted blue bar up and down we can determine the number of clusters to be kept. The placement of the line on a specific level expresses the cut-off value, i.e. the number of the remaining clusters and visualizes the iterative process of hierarchical clustering. Otherwise we can define the maximum number of clusters by clicking on the value we wish to keep from a predefined series of values. When we cannot compute the optimal cut-off value from the produced dendrogram, the system automatically computes the optimal number of clusters to be kept.
- **Relationship maps.** The clustering results are represented into a map that visualizes the relations between clusters (i.e. technological sectors). Every cluster is a bubble, while its size indicates the volume of patents inside the cluster. By double-clicking a bubble, the list of patents that belong to it appears. We can move the bubbles of the map by left-clicking and dragging them anywhere in the screen and thus explore the relationship between the clusters. Furthermore, the width of the lines connecting the bubbles expresses the relationship (similarity) among clusters.

Indicators

Advanced visualization techniques allow the indicators to be explored from many different perspectives. These views can have many forms including bar graphs, plots, trees, spreadsheets, and summary reports.

A first indicator is the one that gives the number of patents that correspond to each area of technology. This permits to identify the maturity of a specific area of technology, as well as innovations. According to our interests, we can obtain either the top areas of technology or the less active technologies based on the number of patents.

Another indicator relates to the level of continents, countries or designated states. The number of patents taken in a given country, broken down by country of the patentee (the inventor or the applicant) or by priority country (first country where the invention is filed

Figure 6. Interactive selection of number of clusters

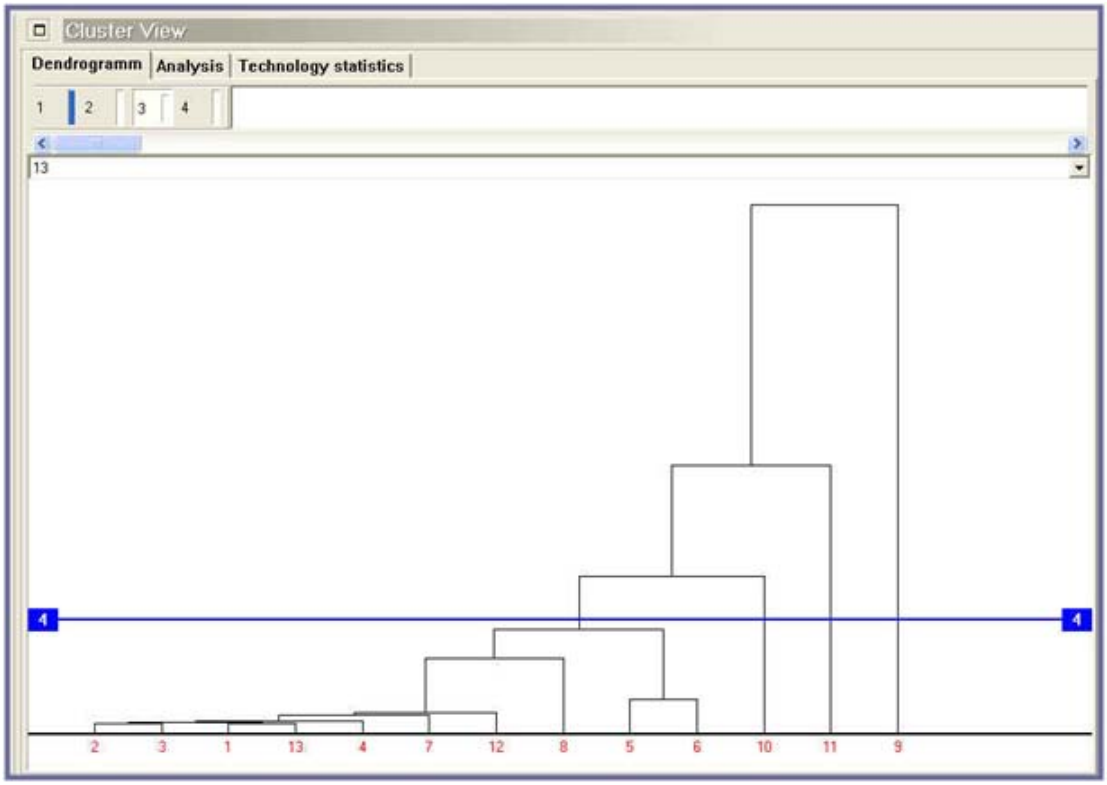
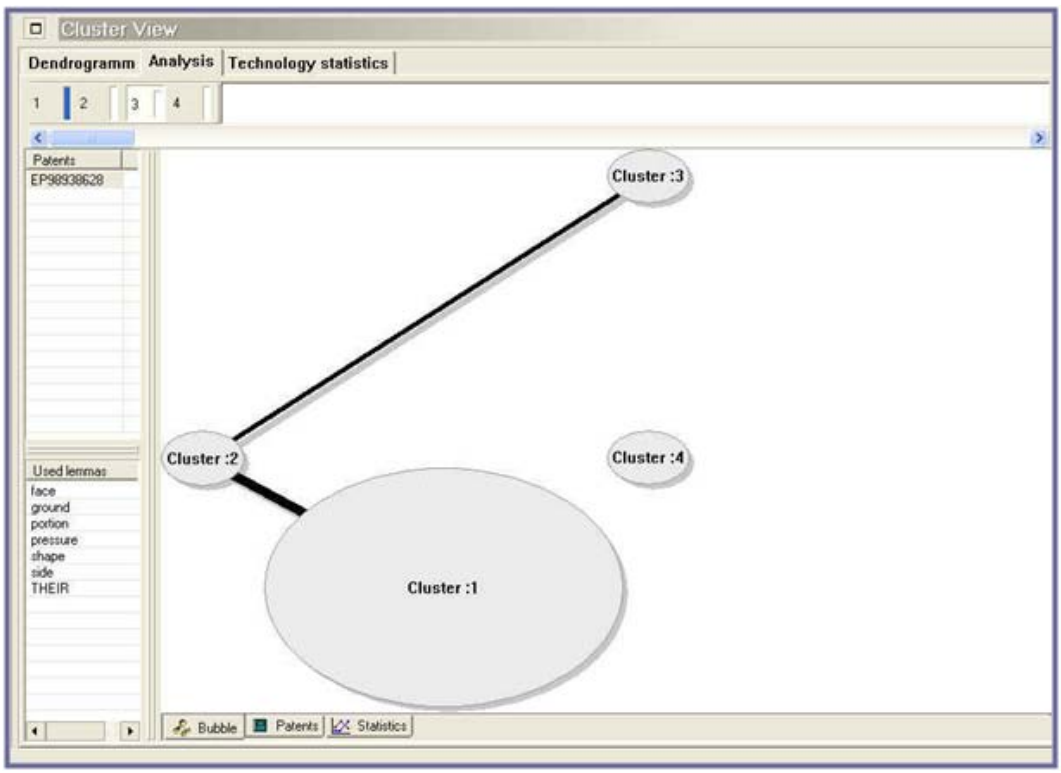


Figure 7. Relationship map of clusters



before protection is extended to other countries) can reveal to what extent do countries share their technological output (i.e. protection for smaller inventions is searched on the local market only). In addition, another indicator is this that gives the top areas of technology of each continent or the less active areas of technology for each continent.

The field of assignees can also be used for producing different kind of indicators in order to catch points of interest. Firstly, the top assignees as well as the less active are of great importance in order to identify the leaders in specific areas of technology. We should have in mind that those who apply a patent vary from individuals to companies public or private, universities, etc. Therefore we should be able to define all these categories that will form the basis for different analyses.

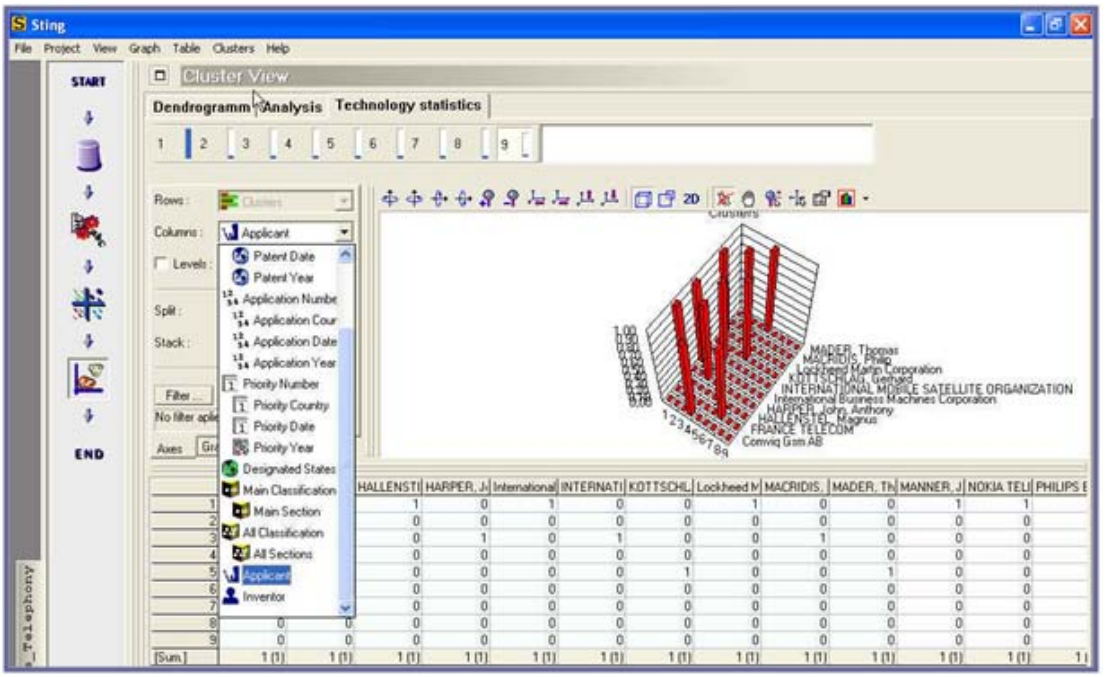
Taking into account the indicators mentioned above, we could derive statistical measures based on the number or percentages of patents for each technology area, as well as for the designated states or inventors. The evolution over time is another feature that helps tracing the technology on-goings and monitoring the novelty of an invention.

FUTURE TRENDS

The previous sections discuss our methodology for patent mining and the tool we have built to experiment with it. Currently, we are exploring the way Semantic Web Mining techniques can be integrated into the model. Especially, ontologies appear as a promising technology since they provide a level of semantics that we do not currently address, allowing improved taxonomies and reasoning about the data and text. The semantic analysis of patents will help in the creation of their richer representation and in the capture of their "hidden" relations and associations. Moreover, it can play significant role in the stage of knowledge discovery so that it can leads to better results. Although these techniques require important calculating time, the investigation of the way that they could be more efficient and scalable particularly for big sets of patents it comprise an important research challenge.

Another area of future research that we believe is promising relates to the problem of multilingualism. The adopted TM techniques were applied in the English texts of patents. It constitutes important question that it requires more study the implementation of algorithms that will process multilingual texts and will produce intermediary forms that do not depend on the language.

Figure 8. Key players in the area of "Mobile Telephony"



The application of mining analysis on these forms will allow the access of new knowledge and will open new possibilities of their exploitation.

Current systems and tools that apply TM techniques address specialized users and experts. Future applications would be useful to incorporate a higher degree of usability so that they concern more users e.g. automatic translation of natural language questions that will execute the suitable operations of mining. Also these tools could have the form of intelligent personal agents which will learn the profile of each user and based on this will execute the TM operations automatically without being necessary the explicit requirement from the user. Besides, the provision of personalized experiences has gained substantial momentum with the rise of Web.

Finally, special interest would present the application of the proposed methodological approach in different types of semi-structured or unstructured data such as documents or web pages. Particularly, its application in Web data e.g. in the problem of web site content classification or the forecast which pages will have the bigger demand we believe that it will lead to useful conclusions.

CONCLUSION

In this article we show that patent data can provide valuable insights for improving the quality of business decisions. The proposed methodological approach analyzes patents based on multidimensional techniques. In this way it facilitates the capture of knowledge not only in the level of a technological sector but also in the level of country or a set of sectors allowing the easy comparison of results and leading to the production of indicators for technological trends. The innovative features of the approach are related mainly to the use and the combination of statistical techniques that allows the more effective analysis of patents.

Firstly, the data used by the tool take into consideration all the information that describes a patent. Compared to the existing approaches (as EUROSTAT), that exploit only the first digits of IPC codes and consequently lose certain information, our tool analyses are based on the exploitation of all fields. At the same time, the use of texts that describes a patent as the title and the abstract allows a more completed description of its technical characteristics. The use of other fields such as countries, businesses, years, etc. helps considerably in the implementation of different types of analyses and in the depiction of information via different ways.

The approach has the possibility to analyze the scientific and technological innovations and the progress in Europe in three different levels. Initially the analysis can be executed in the level of sector. This means that a specific sector can be isolated so that it is analyzed and is determined consequently its technological evolution, which is not obvious. We use for the analysis the code IPC and/or the text that describes the patent so that we create homogeneous clusters from which they can be determined technological tendencies and be drawn useful conclusions. The variables that are not included in the main part of analysis e.g. the businesses that submit the patents, the inventors, the countries in which were submitted the patents, etc. enrich further the result of clustering and they are used as additional variables in the analysis. The same analysis can be executed in the level of all sectors. This allows us to take information on the scientific and technological activity per selected sector with regard to the total of enterprises that submit patents in this particular sector, for a specific year or for few years. Moreover, the analysis in the level of country is available. Through this, homogeneous clusters of countries can be created that present the progress the scientific and technological evolution in the different countries of Europe.

Finally, the innovative feature from the point of used statistical methods is not related so much to the methods themselves but to their combination for the patents' analysis. The incorporation of TM techniques on the processing patents content (titles and abstracts) is one of the most important characteristics of the methodology. The idea is that the vocabulary that is characteristic for the patents categories based on various descriptive variables that are linked with them (as the country or the date of application), it provides additional interesting ideas for the analysis of patents. Competitive indicators with regard to competitive level of each country can be exported, as well as countries that are active in the particular technological region, etc. The combination of correspondence and cluster analysis provides a powerful tool for the efficient exploitation of patents. The production of technology indicators is the most important feature of the proposed methodology.

REFERENCES

Beaugrande, R., & Dressler, W. (1981). *Introduction to Text Linguistics*. London Longman.

Benzecri, J.P. (1992). *Correspondence Analysis Handbook*. New York: Marcel Dekker.

Chappelier, J., Peristera, V., Rajman, M., & Seydoux, F. (2002). Evaluation of Statistical and Technological Innovation Using Statistical Analysis of Patents, *International Conference on the Statistical Analysis of Textual Data (JADT)*.

EPO, European Patent Office (2006). Available at: http://www.european-patent-office.org.

Hotho, A., Nurnberger, A., & Paab, G. (2005). A Brief Survey of Text Mining.

Huang, X. (2006). Clustering Analysis and Algorithms. *Encyclopedia of Data Warehousing and Mining*. Idea Group Inc., 159-164.

Markellos, K., Markellou, P., Mayritsakis, G., Panagopoulou, G., Perdikouri, K., Sirmakessis, S., & Tsakalidis, A. (2003). STING: Evaluation of Scientific and Technological Innovation and Progress in Europe through Patents, *Statistical Data Mining and Knowledge Discovery*, Chapman & Hall/CRC Press.

Narin, F. (1995). Patents as Indicators for the Evaluation of Industrial Research Output, *Scientometrics*, 34(3), 489-496.

Schmoch, U., Bierhals, R., & Rangnow, R. (1998). Impact of International Patent Applications on Patent Indicators. *Joint NESTI/TIP/GSS Workshop*.

KEY TERMS

Business Intelligence (BI): BI is the process of gathering high-quality and meaningful information about the subject being researched to answer questions and identify significant trends or patterns, giving key stakeholders the ability to draw conclusions and make strategic and long-term business decisions.

Cluster Analysis: The process of grouping objects into clusters so that objects are similar to one another within a cluster but dissimilar to objects in other clusters.

Correspondence Analysis: A multivariate method for exploring cross-classified data by finding low dimensional geometrical representations and related numerical statistics. It can reveal features in the data without assuming any underlying distributions of the data.

Linguistic Preprocessing: The process of restricting the morphologic variation of the textual data by reducing each of the different inflections of a given word form to a unique canonical representation (or lemma). It requires a disambiguation, involving a morpho-syntactic analysis and some times a pragmatic analysis.

Patent Databases: Databases that include information about patents and can be distinguished to online ones e.g. WPI(L), EPAT, US, CLAIMS, etc. and those that offered in CD-ROM e.g. ESPACE ACCESS, ESPACE-FIRST, ESPACE BULLETIN, etc.

Patent Mining: The process of searching for meaningful trends, patterns, and relationships among patent records contained in patent databases.

Technology Watch: The activity of surveying the development of new technologies, of new products, of tendencies of technology, as well as measuring their impact on actual technologies, organizations or people. Technology Watch is closely related to innovation.

Text Mining (TM): TM constitutes an extension of traditional Data Mining and concerns the extraction of new, previously unknown and useful information from large volumes of text (semi-structured or unstructured).

Text Mining Methods for Hierarchical Document Indexing

Han-Joon Kim
The University of Seoul, Korea

INTRODUCTION

We have recently seen a tremendous growth in the volume of online text documents from networked resources such as the Internet, digital libraries, and company-wide intranets. One of the most common and successful methods of organizing such huge amounts of documents is to hierarchically categorize documents according to topic (Agrawal, Bayardo & Srikant, 2000; Kim & Lee, 2003). The documents indexed according to a hierarchical structure (termed 'topic hierarchy' or 'taxonomy') are kept in internal categories as well as in leaf categories, in the sense that documents at a lower category have increasing specificity. Through the use of a topic hierarchy, users can quickly navigate to any portion of a document collection without being overwhelmed by a large document space. As is evident from the popularity of web directories such as Yahoo (http://www.yahoo.com/) and Open Directory Project (http://www.dmoz.org/), topic hierarchies have increased in importance as a tool for organizing or browsing a large volume of electronic text documents.

Currently, the topic hierarchies maintained by most information systems are manually constructed and maintained by human editors. The topic hierarchy should be continuously subdivided to cope with the high rate of increase in the number of electronic documents. For example, the topic hierarchy of the Open Directory Project has now reached about 590,000 categories. However, manually maintaining the hierarchical structure incurs several problems. First, such a manual task is prohibitively costly as well as time-consuming. Until now, large search portals such as Yahoo have invested significant time and money into maintaining their taxonomy, but obviously they will not be able to keep up with the pace of growth and change in electronic documents through such manual activity. Moreover, for a dynamic networked resource (e.g., World Wide Web) that contains highly heterogeneous documents accompanied by frequent content changes, maintaining a 'good' hierarchy is fraught with difficulty, and oftentimes is beyond the human experts' capabilities. Lastly, since human editors' categorization decision is not only highly subjective but their subjectivity is also variable over time, it is difficult to maintain a reliable and consistent hierarchical structure. The above limitations require information systems that can provide intelligent organization capabilities with topic hierarchies. Related commercial systems include Verity Knowledge Organizer (http://www.verity.com/), Inktomi Directory Engine (http://www.inktomi.com/), and Inxight Categorizer (http://www.inxight.com/), which enable a browsable web directory to be automatically built. However, these systems did not address the (semi-)automatic evolving capabilities of organizational schemes and classification models at all. This is one of the reasons why the commercial taxonomy-based services do not tend to be as popular as their manually constructed counterparts, such as Yahoo.

BACKGROUND

In future systems, it will be necessary for users to be able to easily manipulate the hierarchical structure and the placement of documents within it (Aggarwal, Gates & Yu, 1999; Agrawal, Bayardo & Srikant, 2000). In this regard, this section presents three critical requirements for intelligent taxonomy construction, and taxonomy construction process using text-mining techniques.

Requirements for Intelligent Taxonomy Construction

1. **Automated classification of text documents:** In order to organize a huge number of documents, it is essential to automatically assign incoming documents to an appropriate location on a predefined taxonomy. Recent approaches towards automated classification have used a supervised

machine-learning paradigm to inductively build a classification model of pre-defined categories from a training set of labeled (pre-classified) data. Basically, such machine-learning based classification requires sufficiently large number of labeled training examples to build an accurate classification model. Assigning class labels to unlabeled documents should be performed by human labeler, and the task is a highly time-consuming and expensive. In fact, it is not easy to obtain a large number of good quality labeled documents in real world operational environments. Thus one of important issues is to develop a reasonable classification model only with insufficient training examples. We must assume that rather than trying to prepare a perfect set of training examples at first, new training documents are continuously provided for learning as a data stream; this means that continuous (on-line) update of the current classification model is required whenever a new set of training examples are prepared.

2. **Semiautomatic management of evolving taxonomy:** The taxonomy initially constructed should change and adapt as its document collection continuously grows or users' needs change. When concept drift happens in particular categories, or when the established criterion for classification alters with time as the content of the document collection changes, it should be possible for part of taxonomy to be reorganized; the system is expected to recommend users a number of feasible sub-taxonomies for the reorganized part.
3. **Making use of domain (or human) knowledge in cluster analysis for topic discovery:** In order to refine the taxonomy, it is necessary to discover new topics (or categories) that can precisely describe the currently indexed document collection. In general, topic discovery is achieved by clustering techniques since clusters that are distinct groups of similar documents can be regarded as representing topically coherent topics in the collection. Clustering for topic discovery is a challenging problem with sufficient domain knowledge. This is because taxonomy should reflect the preferences of an individual user or specific requirements of an application. However, clustering is inherently an unsupervised learning process without depending on external knowledge. Therefore, a new type of supervised clustering is required that reflects external knowledge provided by users.

Taxonomy Construction Process using Text-Mining Techniques

Table 1 illustrates a procedure for hierarchically organizing text documents. The system begins with an initial topic hierarchy in which each document is assigned to its appropriate categories by automatic document classifiers. The topic hierarchy is then made to evolve so as to reflect the current contents and usage of indexed documents. The classification process repeats based on the more refined hierarchy.

Table 1: Procedure for hierarchically organizing text documents

Step 1. Initial construction of taxonomy
i. Define an initial (seed) taxonomy
Step 2. Category (Re-) Learning
i. Collect a set of the controlled training data fit for the defined (or refined) taxonomy
ii. Generate (or Update) the current classification model so as to enable a classification task for newly generated categories
iii. Periodically, update the current classification model so as to constantly guarantee high degree of classification accuracy while refining the training data
Step 3. Automatic Classification
i. Retrieve documents of interest from various sources
ii. Assign each of the unknown documents into more than one categories with its maximal membership value according to the established model
Step 4. Evolution of taxonomy
i. If concept drift or a change in the viewpoint occurs within a sub-taxonomy, reorganize the specified sub-taxonomy
ii. If a new concept sprouts in the unclassified area, perform the cluster analysis for the data within the unclassified area into new categories
Step 5. Sub-taxonomy Construction and Integration
i. Integrate the refined sub-taxonomy or new categories into the main taxonomy
Step 6. Go to Step 2

In Table 1, steps 2 and 3 are related to machine-learning based text classification, step 4 semisupervised clustering for topic discovery, and step 5 taxonomy building.

MAIN THRUST OF THE CHAPTER

This section discusses a series of text mining algorithms that can effectively support the taxonomy construction process. Recent text mining algorithms are prompted by machine learning paradigm; in particular, so are classification and clustering algorithms. Another important issue is about feature selection algorithms because textual data includes a huge number of features such as words or phrases. A feature selection module in the system extracts plain text from each of the retrieved documents and automatically determines only more significant features to speed up the learning process and to improve the classification (or clustering) accuracy. However, this chapter does not present the feature selection algorithms because their related issues are not significantly dependent upon the system.

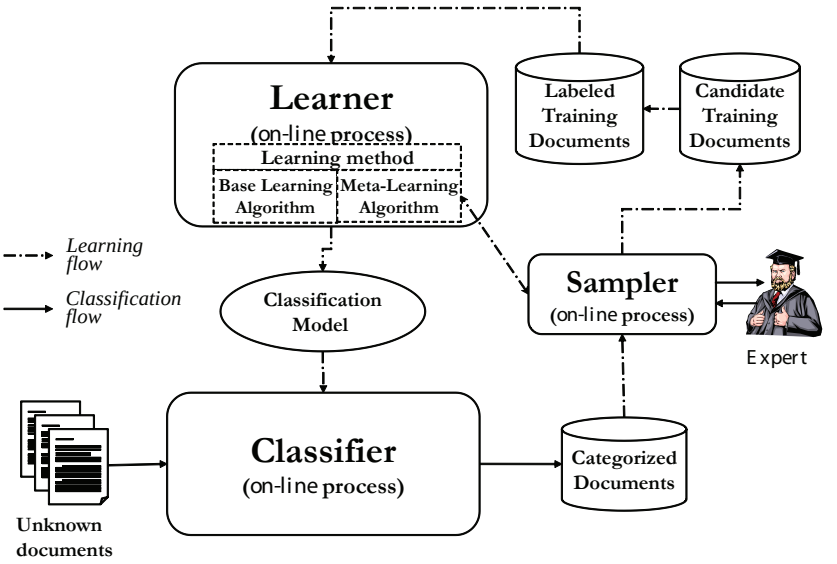
Operational Automated Classification: Combination of Metalearning and On-line Learning

As mentioned before, machine-learning based classification methods require a large number of good quality data for training. However, this requirement is not easily satisfied in real-world operational environments. Recently, many studies on text classification focus on the effective selection of good quality training data that accurately reflect a concept in a given category, rather than algorithm design. How to compose training data has become a very important issue in developing operational classification systems. Recent promising approaches include a combination of 'active learning' and 'semisupervised learning' (Kim & Chang, 2003; Muslea, Minton & Knoblock, 2002) and a combination of 'active learning' and 'boosting' (Kim & Kim, 2004; Liu, Yin, Dong & Ghafoor, 2005)

Firstly, the active learning approach is that the learning module actively chooses the training data from a pool of unlabeled data for humans to give their appropriate class label (Argamon-Engelson & Dagan, 1999). Among different types of active learning, the selective sampling method has been frequently used for learning with text data. It examines a pool of unlabeled data and selects only the most informative ones through a particular measure such as the uncertainty of classification accuracy. Secondly, the semisupervised learning is a variant of supervised learning algorithm in which classifiers can be more precisely learned by augmenting a few labeled training data with many unlabeled data (Demiriz & Bennett, 2000). For semisupervised learning, EM (Expectation-Maximization) algorithm can be used that is an iterative method for finding maximum likelihood in problems with unlabeled data (Dempster, Laird & Rubin, 1977). To develop operational text classifiers, the EM algorithm has been evaluated to be a practical and excellent solution to the problem of lack of training examples in developing classification systems (Nigam, McCallum, Thrun, & Mitchell, 2000). Lastly, boosting is an ensemble learning technique that improves the classification accuracy of supervised learning algorithms by combining a series of base (or weak) classifiers to produce a single powerful classifier (Freund & Schapire, 1996). The Boosting algorithm can significantly improve 'decision tree' learning algorithm (Friedman, Hastie & Tibshirani, 2000) and the 'Naïve Bayes' learning algorithm (Kim HJ, Kim JU & Ra, 2005) with the same amount of training data. Here, the active learning (or selective sampling), EM, and boosting algorithms are included in 'metalearning' algorithms that employ a specific machine learning algorithm (such as Naïve Bayes) as its component.

Figure 1 shows a text classification system architecture, which supports a metalearning process to improve the performance of a text classifier. The system consists of three modules: *Learner*, *Classifier*, and *Sampler*; in contrast, conventional systems do not include the *Sampler* module. The *Learner* module creates a classification model (or function) by examining and analyzing the contents of training documents. The base learning algorithms for text classification can include the Naïve Bayes (Mitchell, 1997b) and Support Vector Machine (Joachims, 2001), and the metalearning algorithms EM and Boosting algorithms. The Naïve Bayes algorithm has been improved by incorporating selective sampling into EM or Boosting algorithm (Kim & Chang, 2003; Kim HJ, Kim JU & Ra, 2005). By using a special measure called 'classification uncertainty' proposed in (Kim & Chang, 2003), new training documents can be automatically selected in the *Sampler* module while interacting with EM or Boosting learning process.

Figure 1: Architecture of an operational classification system



The *Classifier* module uses the classification model built by the *Learner* to determine the categories of each of unknown documents. In the conventional systems, the *Learner* runs only once as an off-line process, but it is desirable to update the current model continuously as an 'on-line' process. To achieve the on-line learning, Naïve Bayes is a good selection. This is because the algorithm can incrementally update the classification model only by adding additional feature estimates to currently learned model instead of re-building the model completely (Yang & Liu, 1999). The learning algorithm has been successfully used for textual data with high dimensional feature space (Agrawal, Bayardo & Srikant, 2000). Moreover, the Naïve Bayes is straightforwardly applied to the EM algorithm due to its probabilistic learning framework (Nigam, McCallum, Thrun, & Mitchell, 2000). Lastly, the *Sampler* module isolates a subset of candidate examples (e.g., through uncertainty-based selective sampling proposed in (Kim & Chang, 2003)) from currently classified data, and returns them to a human expert for class labeling.

The metalearning algorithms assume that a stream of unlabeled documents is provided from some external sources. Practically, rather than acquiring the extra unlabeled data, it is more desirable to use the entire set of data indexed on the current populated taxonomy as a pool of unlabeled documents. As you see in Figure 1, the classified documents are fed into the *Sampler* to augment the current training documents, and then they are used by the *Learner* as a pool of the unlabeled documents for metalearning process. Consequently, in the context of the *Learner* module, not only can we easily obtain the unlabeled data used for metalearning process without extra effort, but also some of the mistakenly classified data are correctly classified.

Semisupervised (User-constrained) Clustering for Topic Discovery

Most clustering algorithms do not allow introducing external knowledge to the clustering process. However, to discover new categories for taxonomy reorganization, it is essential to incorporate external knowledge into cluster analysis. Such a clustering algorithm is called 'semisupervised clustering', which is very helpful in the situation where we should continuously discover new categories from incoming documents. A few strategies for incorporating external human knowledge into cluster analysis have already been proposed in (Talavera & Bejar, 1999; Xing, Ng, Jordan & Russell, 2003). One possible strategy is to vary the distance metrics by weighting dependencies between different components (or features) of feature vectors with the quadratic form distance for similarity scoring. That is, the distance between two document vectors $\mathbf{d}_x$ and $\mathbf{d}_y$ is given by:

$$dist_{\mathbf{W}}(\mathbf{d}_x, \mathbf{d}_y) = \sqrt{(\mathbf{d}_x - \mathbf{d}_y)^{\mathrm{T}} \cdot \mathbf{W} \cdot (\mathbf{d}_x - \mathbf{d}_y)} \qquad (1)$$

where each document is represented as a vector of the form $\mathbf{d}_x = (d_{x1}, d_{x2}, \ldots, d_{xn})$, where n is the total number of index features in the system and $d_{xi} (1 \le i \le n)$ denotes the weighted frequency that feature t_i occurs in document $\mathbf{d}_x$, T denotes the transpose of vectors, and $\mathbf{W}$ is an $n \times n$ symmetrical weight matrix whose entry w_{ij} denotes the interrelationship between the components t_i and t_j of the vectors. Each entry w_{ij} in $\mathbf{W}$ reveals how closely features t_i is associated with feature t_j. If the clustering algorithm uses this type of distance functions, then the clusters reflecting users' viewpoints will be identified more precisely.

To represent user knowledge for topic discovery, one can introduce one or more groups of relevant (or irrelevant) examples to the clustering system, depending on the user's judgment of the selected examples from a given document collection (Kim & Lee, 2002). Each of these document groups is referred to as a 'document bundle', which is divided into two types of bundles: positive and negative ones. Documents within positive bundles (i.e., documents judged jointly 'relevant' by users) must be placed in the same cluster while documents within negative bundles (i.e., documents judged 'irrelevant' by users) must be located in different clusters. Then, when document bundles are given, the clustering process induces the distance metric parameters to satisfy the given bundle constraints. The problem is how to find the weights that best fit the human knowledge represented as document bundles. The distance metric must be adjusted by minimizing the distance between documents within positive bundles that belong to the same cluster while maximizing the distance between documents within negative bundles. This dual optimization problem can be solved using the following objective function $Q(\mathbf{W})$: (see equation (2)).

The above formula is a function of weight matrix $\mathbf{W}$ since the distance value depends on the weight matrix. In the above formula, document bundle set B^+ (or B^-) is defined to be a collection of positive (or negative) bundles, and $\langle \mathbf{d}_x, \mathbf{d}_y \rangle \in R_B+$ or $\langle \mathbf{d}_x, \mathbf{d}_y \rangle \in R_B-$ denotes that a pair of documents $\mathbf{d}_x$ and $\mathbf{d}_y$ is found in positive bundles or negative bundles, respectively. Each pair within the bundles is processed as a training example for learning the weighted distance measure, and we should find a weight matrix that minimizes the objective function. The reasons why the real distance value is squashed by θ is that the sum of the distance values of documents within R_{B^-} during the learning process can dominate the total sum of the distance values of documents within $R_{B^+} \cup R_{B^-}$. A squashing function θ is used that converges to $k/2$ and its steepness is determined by σ. To search for an optimal matrix, we can use a gradient descent search method that is used for tuning weights among neurons in artificial neural networks (Mitchell, 1997a).

Each entry w_{ij} in the weight matrix is iteratively updated according to the following training rule:

$$w_{ij} \leftarrow w_{ij} + \Delta w_{ij} = w_{ij} + \left(-\eta \cdot \frac{\partial Q}{\partial w_{ij}} \right) \qquad (3)$$

Equation (2).

$$Q(\mathbf{W}) = \sum_{(\mathbf{d}_x, \mathbf{d}_y) \in R_{B^+} \cup R_{B^-}} I(\mathbf{d}_x, \mathbf{d}_y) \cdot \theta\left(dist_{\mathbf{W}}(\mathbf{d}_x, \mathbf{d}_y)\right)$$

$$I(\mathbf{d}_x, \mathbf{d}_y) = \begin{cases} +1 \ if < \mathbf{d}_x, \mathbf{d}_y > \in R_{B^+} \\ -1 \ if < \mathbf{d}_x, \mathbf{d}_y > \in R_{B^-} \end{cases}$$

$$\theta(z) = \frac{k}{1 + e^{\frac{\sigma}{z}}}$$

where

$$R_{B^+} = \left\{ \left\langle \mathbf{d}_x, \mathbf{d}_y \right\rangle | \, \mathbf{d}_x \in B^+ \ and \ \mathbf{d}_y \in B^+ \ for \ any \ positive \ bundle \ set \ B^+ \right\}$$

$$R_{B^-} = \left\{ \left\langle \mathbf{d}_x, \mathbf{d}_y \right\rangle | \, \mathbf{d}_x \in B^- \ and \ \mathbf{d}_y \in B^- \ for \ any \ negative \ bundle \ set \ B^- \right\}$$

$$\frac{\partial Q}{\partial w_{ij}} = \sum_{\langle \mathbf{d}_x, \mathbf{d}_y \rangle \in R_{B^+} \cup R_{B^-}} I(\mathbf{d}_x, \mathbf{d}_y) \cdot \frac{k\sigma e^{\frac{\sigma}{z}} (d_{xi} - d_{yi})(d_{xj} - d_{yj})}{2z^3 (1 + e^{\frac{\sigma}{z}})^2} \quad (4)$$

where $z - dist_w(\mathbf{d}_x, \mathbf{d}_y)$, η is a positive constant called the 'learning rate', which controls the amount of weight adjustment at each step of training.

When concept drift or a change in a user's viewpoint occurs within a particular area of taxonomy, the user should prepare a set of document bundles as external knowledge reflecting the concept drift or the change in viewpoint. Then, based on the prepared user constraint, the clustering process discovers categories resolving the concept drift or reflecting changes in user's viewpoint, and then the isolated categories are incorporated into the main taxonomy.

Automatic Taxonomy Construction

For building hierarchical relationships among categories, we need to note that a category is represented by topical terms reflecting its concept (Sanderson & Croft, 1999). This suggests that the relations between categories can be determined by describing the relations between their significant terms. In this regard, we find that it is difficult to dichotomize the relations between categories into groups representing the presence or absence of association, because term associations are generally represented not as crisp relations, but as probabilistic equations. Thus, degrees of association between two categories can be represented by membership grade in a fuzzy (binary) relation. That is, the generality and specificity of categories can be expressed by aggregating the relations among their terms. In (Kim & Lee, 2003), a hierarchical relationship between two categories is represented by membership grade in a fuzzy (binary) relation. The fuzzy relation $CSR(c_i, c_j)$ (which represents the relational concept "c_i *subsumes* c_j"), called 'category subsumption relation' (*CSR*), between two categories c_i and c_j is defined as follows:

$$\mu_{CSR}(c_i, c_j) = \frac{\sum_{t_i \in V_{c_i}, t_j \in V_{c_j}, \Pr(t_i|t_j) > \Pr(t_j|t_i)} \tau_{c_i}(t_i) \times \tau_{c_j}(t_j) \times \Pr(t_i \mid t_j)}{\sum_{t_i \in V_{c_i}, t_j \in V_{c_j}} \tau_{c_i}(t_i) \times \tau_{c_j}(t_j)} \quad (5)$$

where $\tau_c(t)$ denotes the degree to which the term t represents the concept corresponding to the category c, which can be estimated by calculating the χ^2 statistic value of term t in category c since the χ^2 value represents the degree of term importance (Yang & Pedersen, 1997). $\Pr(t_i|t_j)$ should be weighted by the degree of significance of the terms t_i and t_j in their categories, and thus the membership function $\mu_{CSR}(\cdot)$ for categories is calculated as the weighted average of the values of $\Pr(t_i|t_j)$ for terms. The function value of $\mu_{CSR}(\cdot)$ is represented by a real number in the closed interval [0,1], and indicates the strength of the relationship present between two categories. By using the above fuzzy relation, we can build a sub-taxonomy of categories discovered by cluster analysis. A procedure for building a topic hierarchy is the following:

- **Step 1:** Calculate the *CSR* matrix with entries representing the degree of membership in a fuzzy relation *CSR* for a given set of categories (see Eq. (5)).
- **Step 2:** Generate the α-cut matrix of the *CSR* matrix (denoted by CSR_α) by determining an appropriate value of α.
- **Step 3:** Create a hierarchy of the partitioned categories from the CSR_α matrix representing partial ordering.

FUTURE TRENDS

In applying text-mining techniques to hierarchically organizing large textual data, a number of issues remain to be explored in the future. A practical issue in machine-learning based text classification is how to continuously update the current classification model with insufficient training data while achieving its high accuracy. A good approach to this issue is to use metalearning algorithms such as EM and Boosting. Additionally, an important issue related to topic discovery is how to seamlessly reflect human knowledge to text mining algorithms. Recent studies show that the semisupervised clustering technique that employs document bundle constraint can effectively enhance the cluster quality. As an example, for semisupervised clustering, (Kim & Lee, 2002) attempted to generate external human knowledge of bundle constraints through user-relevance feedback. And (Basu, Bilenko, & Mooney, 2004) proposed a probabilistic framework for semisupervised clustering. In other different aspects, the problem will continue to be intensively tackled.

In terms of automatic taxonomy construction, semantically richer information needs to be automatically built beyond the subsumption hierarchy information of categories; for example, relevance relationship among categories needs to be extracted for cross-referencing of categories. Another challenging issue is that for extracting more precise document context, it is necessary to utilize structural and contextual features (e.g., tree-like structures and diverse tag information of XML documents) of the original textual data, if any, as well as the simple features of 'a bag of words'. In fact, such feature engineering is more profitable and effective than algorithm design, particularly in building commercial applications.

On a practical level, an open challenge is automatic taxonomy engineering for manually constructed topic hierarchies such as Yahoo directory (http://www.yahoo.com/), ODP directory (http://www.dmoz.org/) and UNSPSC classification system (http://www.unspsc.org/). Since these topic hierarchies are popular and huge in size, they are expected to be good exemplars to evaluate practical value of text-mining techniques for taxonomy building.

CONCLUSION

Towards intelligent taxonomy engineering for large textual data, text-mining techniques are of great importance. In this chapter, for developing operational classification systems, a employment of metalearning algorithms such as selective sampling, EM, and boosting algorithms has been introduced together with the classification system architecture that has the on-line learning framework. In terms of category discovery, a simple representation, called document bundles, of human knowledge has been discussed as a way of incorporating human knowledge into cluster analysis for semisupervised clustering. As for taxonomy building, the simple fuzzy-relation based algorithm is described without any complicated linguistic analysis.

The current research on building hierarchical structure automatically is still in an early stage. Basically, current techniques consider only subsumption hierarchy, but future studies should try to extract other useful semantics of discovered categories. More importantly, how to incorporate human knowledge into text-mining algorithm should be further studied with user interaction design. In this regard, semisupervised clustering, semisupervised learning, and metalearning are challenging issues with both academic and practical values.

REFERENCES

Aggarwal, C.C., Gates, S.C., & Yu, P.S. (1999, August). On the Merits of Building Categorization Systems by Supervised Clustering. *International Conference on Knowledge Discovery and Data Mining, KDD'99*. San Diego, USA, 352-356.

Agrawal, R., Bayardo, R., & Srikant, R. (2000, March). Athena: Mining-based Interactive Management of Text Databases. *International Conference on Extending Database Technology, EDBT-2000*. Konstanz, Germany, 365-379.

Argamon-Engelson, S., & Dagan, I. (1999). Committee-Based Sample Selection for Probabilistic Classifiers. *Journal of Artificial Intelligence Research*, 11, 335-360.

Basu, S., Bilenko, M., & Mooney, R.J. (2004, August). A Probabilistic Framework for Semi-supervised Clustering. *International conference on Knowledge discovery and data mining, KDD 2004*. Seattle, USA, 59-68.

Demiriz, A., & Bennett, K. (2000). Optimization Approaches to Semi-Supervised Learning. In M. Ferris, O. Mangasarian, and J. Pang (Eds.), *Applications and Algorithms of Complementarity*, Boston: Kluwer Academic Publishers.

Dempster, A.P., Laird, N., & Rubin, D.B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society*, B39, 1-38.

Freund, Y. & Schapire, R.E. (1996, July). Experiments with a New Boosting Algorithm, The International Conference on Machine Learning, ICML'96, Bari, Italy, 148-156.

Friedman, J.H., Hastie, T., & Tibshirani, R. (2000). Additive Logistic Regression: A statistical view of Boosting, *Annals of Statistics*, 28 (2), 337-374.

Joachims, T. (2001, September). A Statistical Learning Model of Text Classification with Support Vector Machines. *International Conference on Research and*

Development in Information Retrieval, SIGIR-2001. New Orleans, USA, 128-136.

Kim, H.J., & Chang, J.Y. (2003, October). Improving Naïve Bayes Text Classifier with Modified EM Algorithm. *Lecture Notes on Aritificial Intelligence*, 2871, 326-333.

Kim, H.J., Kim, J.U., & Ra, Y.K. (2005) Boosting Naive Bayes Text Classification using Uncertainty-based Selective Sampling. *Neurocomputing* , 67, 403-410.

Kim, H.J., & Lee, S.G. (2002). User Feedback-Driven Document Clustering Technique for Information Organization. *IEICE transactions on Information and Systems*, E85-D(6), 1043-1048.

Kim, H.J., & Lee, S.G. (2003). Building Topic Hierarchy based on Fuzzy Relations. *Neurocomputing*, 51, 481-486.

Liu, X., Yin, J., Dong, J., & Ghafoor, M.A. (2005, October). An Improved FloatBoost Algorithm for Naïve Bayes Text Classification. *International Conference on Advances in Web-Age Information Management, WAIM 2005*, Hangzhou, China, 162-171.

Mitchell, T.M. (1997a). *Artificial Neural Networks*: *Machine Learning*. New York: McGraw-Hill.

Mitchell, T.M. (1997b). *Bayesian Learning*: *Machine Learning*. New York: McGraw-Hill.

Muslea, I., Minton, S., & Knoblock, C. (2002, July). Active + semi-supervised learning = robust multi-view learning. *International Conference on Machine Learning, ICML-2002*. Sydney, Australia, 435-442.

Nigam, K., McCallum, A., Thrun S., & Mitchell, T.M. (2000). Text Classification from Labeled and Unlabeled Documents using EM. *Machine Learning*, 39(2/3), 103-134.

Sanderson, M., & Croft, B. (1999, August). Deriving Concept Hierarchies from Text. *International Conference on Research and Development in Information Retrieval, SIGIR'99*. Berkeley, USA, 206-213.

Talavera, L., & Bejar, J. (1999, August) Integrating Declarative Knowledge in Hierarchical Clustering Tasks. *International Conference on Intelligent Data Analysis, IDA'99*. Amsterdam, The Netherlands, 211-222.

Xing, E.P., Ng, A.Y., Jordan, M.I., & Russell, S. (2002, December). Distance Metric Learning with Application to Clustering with Side-information, *Neural Information Processing Systems, NIPS 2002*. Vancouver, Canada, 505-512.

Yang, Y., & Liu, X. (1999, August). A Re-examination of Text Categorization Methods. *International Conference on Research and Development in Information Retrieval, SIGIR'99*. Berkeley, USA, 42-49.

Yang, Y., & Pedersen, J.O. (1997, July). A Comparative Study on Feature Selection in Text Categorization. *International Conference of Machine Learning, ICML'97*. Nashville, USA, 412-420.

KEY TERMS

Active Learning: Learning modules that support active learning select the best examples for class labeling and training without depending on a teacher's decision or random sampling.

Boosting: A machine learning metaalgorithm for performing supervised learning in which a set of weak learners create a single strong learner.

Concept Drift: A temporal phenomenon in which the general subject matter of information within a category may no longer suit the subject that best explained that information when it was originally created.

Document Clustering: An unsupervised learning technique that partitions a given set of documents into distinct groups of similar documents based on similarity or distance measures.

EM Algorithm: An iterative method for estimating maximum likelihood in problems with incomplete (or unlabeled) data. EM algorithm can be used for semi-supervised learning (see below) since it is a form of clustering algorithm that clusters the unlabeled data around the labeled data.

Fuzzy Relation: In fuzzy relations, degrees of association between objects are represented not as crisp relations but membership grade in the same manner as degrees of set membership are represented in a fuzzy set.

Metalearning Algorithm: a subfield of machine learning where machine learning algorithms are applied on metadata such as properties of the learning problem,

performance measures, or patterns previously derived from the data in order to improve the performance of existing machine learning algorithms.

Supervised Learning: A machine learning technique for inductively building a classification model (or function) of a given set of classes from a set of training (pre-labeled) examples

Semisupervised Clustering: A variant of unsupervised clustering techniques without requiring external knowledge. Semi-supervised clustering performs clustering process under various kinds of user constraints or domain knowledge.

Semisupervised Learning: A variant of supervised learning that uses both labeled data and unlabeled data for training. Semi-supervised learning attempts to provide more precisely learned classification model by augmenting labeled training examples with information exploited from unlabeled data.

Text Classification: The task of automatically assigning a set of text documents to a set of predefined classes. Recent text classification methods adopt supervised learning algorithms such as Naïve Bayes and support vector machine.

Topic Hierarchy (Taxonomy): Topic hierarchy in this chapter is a formal hierarchical structure for orderly classification of textual information. It hierarchically categorizes incoming documents according to topic in the sense that documents at a lower category have increasing specificity.

Theory and Practice of Expectation Maximization (EM) Algorithm

Chandan K. Reddy
Wayne State University, USA

Bala Rajaratnam
Stanford University, USA

INTRODUCTION

In the field of statistical data mining, the Expectation Maximization (EM) algorithm is one of the most popular methods used for solving parameter estimation problems in the maximum likelihood (ML) framework. Compared to traditional methods such as steepest descent, conjugate gradient, or Newton-Raphson, which are often too complicated to use in solving these problems, EM has become a popular method because it takes advantage of some problem specific properties (Xu et al., 1996). The EM algorithm converges to the local maximum of the log-likelihood function under very general conditions (Demspter et al., 1977; Redner et al., 1984). Efficiently maximizing the likelihood by augmenting it with latent variables and guarantees of convergence are some of the important hallmarks of the EM algorithm.

EM based methods have been applied successfully to solve a wide range of problems that arise in fields of pattern recognition, clustering, information retrieval, computer vision, bioinformatics (Reddy et al., 2006; Carson et al., 2002; Nigam et al., 2000), etc. Given an initial set of parameters, the EM algorithm can be implemented to compute parameter estimates that locally maximize the likelihood function of the data. In spite of its strong theoretical foundations, its wide applicability and important usage in solving some real-world problems, the standard EM algorithm suffers from certain fundamental drawbacks when used in practical settings. Some of the main difficulties of using the EM algorithm on a general log-likelihood surface are as follows (Reddy et al., 2008):

- EM algorithm for mixture modeling converges to a local maximum of the log-likelihood function very quickly.
- There are many other promising local optimal solutions in the close vicinity of the solutions obtained from the methods that provide good initial guesses of the solution.
- Model selection criterion usually assumes that the global optimal solution of the log-likelihood function can be obtained. However, achieving this is computationally intractable.
- Some regions in the search space do not contain any promising solutions. The promising and non-promising regions co-exist and it becomes challenging to avoid wasting computational resources to search in non-promising regions.

Of all the concerns mentioned above, the fact that most of the local maxima are not distributed uniformly makes it important to develop algorithms that not only help in avoiding some inefficient search over the low-likelihood regions but also emphasize the importance of exploring promising subspaces more thoroughly (Zhang et al, 2004). This subspace search will also be useful for making the solution less sensitive to the initial set of parameters. In this chapter, we will discuss the theoretical aspects of the EM algorithm and demonstrate its use in obtaining the optimal estimates of the parameters for mixture models. We will also discuss some of the practical concerns of using the EM algorithm and present a few results on the performance of various algorithms that try to address these problems.

BACKGROUND

Because of its greedy nature, the EM algorithm converges to a local maximum on the log-likelihood surface. Hence, the final solution will be very sensitive to the given initial set of parameters. This local maxima problem (popularly known as the initializa-

tion problem) is one of the well studied issues in the context of the EM algorithm. Several algorithms have been proposed in the literature to try and solve this issue (Reddy, 2007).

Although EM and its variants have been extensively used in the literature, several researchers have approached the problem by identifying new techniques that give good initialization. More generic techniques like deterministic annealing (Ueda et al., 1998), genetic algorithms (Pernkopf et al., 2005) have been successfully applied to obtain good parameter estimates. Though, these techniques have asymptotic guarantees, they are very time consuming and hence cannot be used in most practical applications. Some problem specific algorithms like split and merge EM (Ueda et al., 2000), component-wise EM (Figueiredo et al., 2002), greedy learning (Verbeek et al., 2003), parameter space grid (Li, 1999) have also been proposed in the literature. Some of these algorithms are either computationally very expensive or infeasible when learning mixture models in high dimensional spaces (Li, 1999). In spite of the high computational cost associated with these methods, very little effort has been taken to explore promising subspaces within the larger parameter space. Most of the above mentioned algorithms eventually apply the EM algorithm to move to a locally maximal set of parameters on the log-likelihood surface. Simpler practical approaches like running EM from several random initializations, and then choosing the final estimate that leads to the local maximum with the highest log-likelihood value to a certain extent have also been successful.

For a problem with a non-uniform distribution of local maxima, it is difficult for most methods to search neighboring subspaces (Zhang et al, 2004). Though some of these methods apply other additional mechanisms (like perturbations) to escape out of local optimal solutions, systematic methods for searching the subspace have not been thoroughly studied. More recently, TRUST-TECH based Expectation Maximization (TRUST-TECH-EM) algorithm has been developed by Reddy et al (2008), which applies some properties of the dynamical system of the log-likelihood surface to identify promising initial starts for the EM algorithm. This dynamical system approach will reveal more information about the neighborhood regions and helps in moving to different basins of attraction in the neighborhood of the current local maximum.

MAIN FOCUS

In this section, we will first discuss the theoretical aspects of the EM algorithm and prove some of its basic properties. We will then demonstrate the use of the EM algorithm in the context of mixture models and give some comparative results on multiple datasets.

THEORY OF THE EM ALGORITHM

Formally consider the problem of maximizing the likelihood function $L(\theta;x)$ arising from a density $f(x;\theta)$, with x denoting the data or sample, and θ the parameter of interest. As noted above, in both theoretical and applied problems maximizing $L(\theta;x)$ can often be a difficult task. Let us assume that we can identify another random variable y such that

$$f(x;\theta) = \int f(x,y;\theta)dy \qquad (1)$$

and where the likelihood function arising from $f(x,y;\theta)$ is relatively easier to maximize. The variable y is often called the "hidden", "latent" or "missing" data and together (x,y) is often referred to as the "complete" data.

The EM algorithm maximizes the original likelihood function by working with the complete or augmented likelihood. The expectation or E-step takes the expected value of the complete likelihood over the missing data given the original data y and a starting parameter value. This process gives rise to an expected (rather conditional expectation) version of the complete likelihood which is easier to maximize. The E-step essentially has the effect of "substituting" values for the hidden variable y. The maximization or M-step optimizes the resulting conditional expectation of the complete likelihood leading to a new parameter estimate. Based on the new parameter estimate, the E-step and the M-step are repeated back and forth in an iterative manner (McLachlan et al., 1997).

We shall prove below that every EM-step gives an improvement in the likelihood in the original problem but let us first formally state the EM algorithm.

The EM Algorithm

Consider the likelihood function $L(\theta;x)$ and the corresponding log-likelihood $\lambda(\theta;x)$ based on the density $f(x;\theta)$ of the observed data and let $f(x,y;\theta)$ be the density of the augmented data as defined in eqn(1).

Start with an initial value of the parameter estimates given by θ_0 and for a given θ_t $t = 0,1,2,\ldots$ obtain θ_{t+1} by iterating between the following two steps:

E-step - Compute the function $Q(\theta;\theta_t)$ where

$$Q(\theta;\theta_t) = E_{\theta_t}\left[\log \frac{f(x,y;\theta)}{f(x,y;\theta_t)} \middle| X=x\right]$$

Note 1: The quantities θ, θ_t, and x are considered as fixed and the expectation is taken over the hidden variable y.

M-step - Maximize $Q(\theta;\theta_t)$ to obtain the new parameter estimate θ_{t+1} i.e.

$$\theta_{t+1} = \arg\max_{\theta} Q(\theta\ ;\theta_t)$$

Note 2: Here the function $Q(\theta;\theta_t)$ is maximized with respect to θ yielding the updated parameter value θ_{t+1}.

Note 3: Since the denominator in the definition of $Q(\theta;\theta_t)$ does not depend on θ, one could as well maximize the conditional expectation of the log of $f(x,y;\theta)$, i.e. the numerator, w.r.t θ and obtain θ_{t+1}. This is often done in practice as we will see when we apply the EM algorithm to maximum likelihood estimation in Gaussian mixture models.

Basic Properties of EM Algorithm

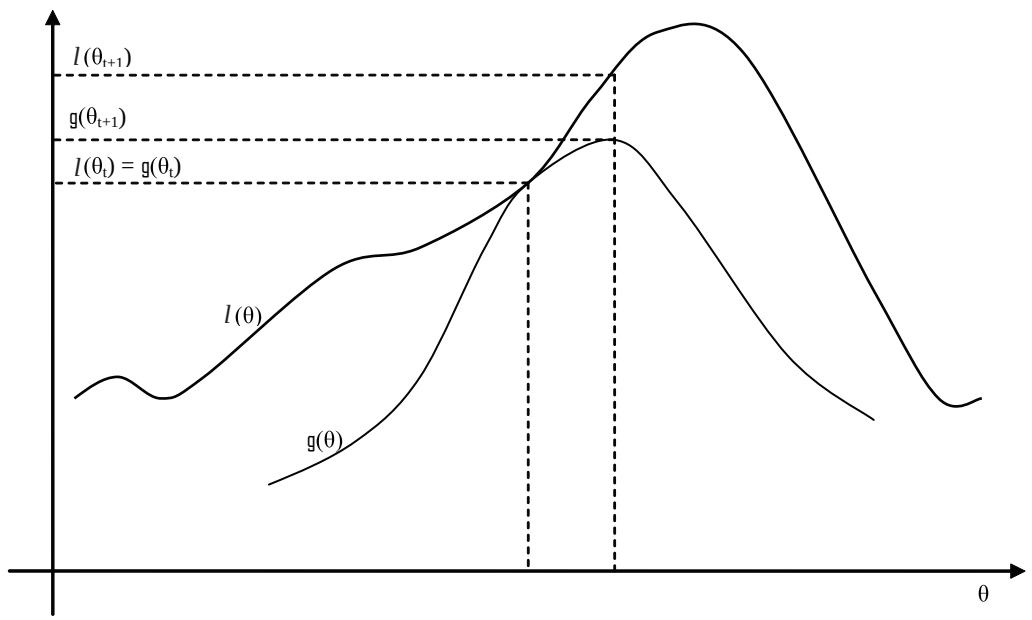
We now proceed to demonstrate that at every EM iteration the likelihood value does not decrease.

Theorem 1: $L(\theta_{t+1}) \geq L(\theta_t)$ for all t [or equivalently, $l(\theta_{t+1}) - l(\theta_t) \geq 0$]

The proof is found in the appendix.

An intuitive interpretation of the working of the EM algorithm stems from noting the following: The E-step introduces a function $g(\theta)$ which is always bounded above by the log likelihood function $l(\theta)$. Moreover, at the current value $\theta = \theta_t$, the original log-likelihood

Figure 1.Illustration of a single EM step. Note that the two functions l(θ) and g(θ) are equal at θ = θ t and that g(θ) is always bounded above by the log likelihood function l(θ). Hence maximizing g(θ) cannot lead to a decrease in the value of l(θ).



function $l(\theta)$ and $g(\theta)$ are equal, i.e. $l(\theta_t) = g(\theta_t)$. The M-step maximizes $g(\theta)$. So at each EM cycle the two functions start off at the same value and since $l(\theta) \geq g(\theta)$, increasing $g(\theta)$ guarantees that the value of $l(\theta)$ also increases. Figure 1 gives a graphical illustration of a single EM step.

CASE STUDY: MIXTURE MODELS

One of the most well-known applications of the EM algorithm is its use in finite mixture models. In this section, we shall give a concrete example where using the EM algorithm can significantly facilitate the task of finding maxima of the likelihood of a finite Gaussian mixture model (GMM) (McLachlan et al., 2000).

Let us assume that there are k components in the mixture model with probability density function given as follows:

$$p(x \mid \Theta) = \sum_{i=1}^{k} \alpha_i \, p(x \mid \theta_i) \tag{4}$$

Where $x=[x_1, x_2,\ldots, x_d]'$ is the feature vector of d dimensions. The α's represent the mixing weights and the Θ represents the parameter set $[\alpha_1, \alpha_2, .., \alpha_k, \theta_1, \theta_2,\ldots,\theta_k]$. Also it should be noted that being probabilities the α_i (with $0 \leq \alpha_i \leq 1$) should sum to 1.

One of the goals of learning in mixture models is to obtain parameter estimates for Θ from a sample of n data points, denoted by $X = \{x^{(1)},x^{(2)},\ldots, x^{(n)}\}$, drawn from a distribution with density given by eqn(4). The maximum likelihood estimate (MLE) is given as follows:

$$\hat{\Theta}_{MLE} = \arg\max_{\Theta}[\log \; p(x \mid \Theta)] \tag{5}$$

Since in general this MLE cannot be found analytically for Gaussian mixture models, one has to rely on iterative procedures that can find the global maximum of the likelihood function. The EM algorithm has been used successfully to find the local maxima of such a function.

The EM Algorithm for Mixture Models

The EM algorithm takes X to be the observed data and the missing part, termed as "hidden data" is a set of n labels $Y = \{y^{(1)},y^{(2)},\ldots, y^{(n)}\}$ associated with each of the n sample points, indicating which component produced each sample (McLachlan et al., 1997). Each label $y^{(j)} = [y_1^{(j)},y_2^{(j)},\ldots, y_k^{(j)}]$ is a binary vector where $y_i^{(j)} = 1$ and $y_m^{(j)} = 0$ for all $m \neq i$, indicating that sample $x^{(j)}$ was produced by the i^{th} component. Now, the complete log-likelihood (i.e. the one from which we would estimate Θ for the complete data $U = \{X, Y\}$) is:

$$\log p(U \mid \Theta) = \sum_{j=1}^{n} \log \prod_{i=1}^{k} [\alpha_i \, p(x^{(j)} \mid \theta_i)]^{y_i^{(j)}} \tag{6}$$

$$\log p(U \mid \Theta) = \sum_{j=1}^{n} \sum_{i=1}^{k} y_i^{(j)} \log \; [\alpha_i \, p(x^{(j)} \mid \theta_i)] \tag{7}$$

The EM algorithm produces a sequence of estimates $\{\hat{\Theta}(t), \;\; t = 0,1,2,\ldots\}$ by alternating between the following two steps until convergence:

- **E-Step:** Compute the conditional expectation of the hidden data, given X and the current estimate $\hat{\Theta}(t)$. Since $\log p\,(X, Y \mid \Theta)$ from eqn(7) is linear with respect to the missing data Y, we simply have to compute the conditional expectation $W = E\,[Y \mid X, \Theta(t)]$ in $\log p\,(X, Y \mid \Theta)$. We can therefore define the Q-function as follows:

$$Q(\Theta \mid \hat{\Theta}(t)) \equiv E_Y[\log \;\; p(X,Y) \mid X, \hat{\Theta}(t)] \tag{8}$$

As Y is a binary vector, its conditional expectation is given by :

$$\begin{aligned} w_i^{(j)} &\equiv E \;\; [y_i^{(j)} \mid X, \hat{\Theta}(t)] \\ &= \Pr \; [y_i^{(j)} = 1 \mid x^{(j)}, \hat{\Theta}(t)] \\ &= \frac{\hat{\alpha}_i(t) \, p(x^{(j)} \mid \hat{\theta}_i(t))}{\sum_{i=1}^{k} \hat{\alpha}_i(t) \, p(x^{(j)} \mid \hat{\theta}_i(t))} \end{aligned} \tag{9}$$

where the last equality follows from Bayes law (α_i is the a priori probability that $y_i^{(j)} = 1$), while $w_i^{(j)}$ is the a posteriori probability that $y_i^{(j)} = 1$ given the observation $x^{(j)}$.

- **M-Step:** The parameter estimates are updated using the following equation:

$$\hat{\Theta}(t+1) = \arg\max_{\Theta}\{Q(\Theta, \hat{\Theta}(t))\} \quad (10)$$

EM for Gaussian Mixture Models

We now proceed to illustrate the workings of the EM algorithm in the context of Gaussian Mixture Models. The convergence properties of the EM algorithm for Gaussian mixtures are thoroughly discussed in (Xu et al., 1996). The function $p(x|\theta_i)$ is a multivariate Gaussian density parameterized by θ_i (i.e. μ_i and Σ_i):

$$p(x|\theta_i) = \frac{|\Sigma_i|^{-1/2}}{(2\pi)^{d/2}} e^{-\frac{1}{2}(x-\mu_i)'\Sigma_i^{-1}(x-\mu_i)}$$

The *Q*-function for Gaussian mixture models (GMMs) is given by:

$$Q(\Theta|\hat{\Theta}(t)) = \sum_{j=1}^{n}\sum_{i=1}^{k} w_i^{(j)}[\log\frac{|\Sigma|^{-\frac{1}{2}}}{(2\pi)^{\frac{d}{2}}} - \frac{1}{2}(x^{(j)}-\mu_i)^T\Sigma_i^{-1}(x^{(j)}-\mu_i) + \log\alpha_i] \quad (11)$$

Where

$$w_i^{(j)} = \frac{\hat{\alpha}_i(t)|\hat{\Sigma}_i(t)|^{-\frac{1}{2}} e^{-\frac{1}{2}(x^{(j)}-\hat{\mu}_i(t))^T|\hat{\Sigma}_i(t)|^{-1}(x^{(j)}-\hat{\mu}_i(t))}}{\sum_{i=1}^{k}\hat{\alpha}_i(t)|\hat{\Sigma}_i(t)|^{-\frac{1}{2}} e^{-\frac{1}{2}(x^{(j)}-\hat{\mu}_i(t))^T|\hat{\Sigma}_i(t)|^{-1}(x^{(j)}-\hat{\mu}_i(t))}} \quad (12)$$

The maximization step is given by the following equation:

$$\frac{\partial}{\partial\Theta_k} Q(\Theta|\hat{\Theta}(t)) = 0 \quad (13)$$

where Θ_k denotes the parameters for the k^{th} component. As the posterior probabilities in the E-step now appear in the *Q*-function as constants, the *Q*-function resembles the Gaussian likelihood when the components are pre-specified (Wasserman, 2004). Maximizing this function in the M-step now becomes trivial as the problem essentially reduces to finding MLEs in a (single-component) Gaussian model, the solution to which is well known and is available in closed form expressions. One can easily verify that the updates for the maximization step in the case of GMMs are given as follows:

$$\mu_i(t+1) = \frac{\sum_{j=1}^{n} w_i^{(j)} x^{(j)}}{\sum_{j=1}^{n} w_i^{(j)}}$$

$$\Sigma_i(t+1) = \frac{\sum_{j=1}^{n} w_i^{(j)}(x^{(j)}-\mu_i(t+1))(x^{(j)}-\mu_i(t+1))^T}{\sum_{j=1}^{n} w_i^{(j)}} \quad (14)$$

$$\alpha_i(t+1) = \frac{1}{n}\sum_{j=1}^{n} w_i^{(j)}$$

Results of Various Global Optimization Methods

Table 1 compares the performance of different methods proposed in the literature such as k-means+EM , split and merge EM (SMEM) and TRUST-TECH-EM for two datasets. RS+EM corresponds to just a single random start and is shown here to illustrate the empirical lower bound on the performance of all these methods. The mean and the standard deviations of the log-likelihood values are reported. Higher log-likelihood values correspond to better parameter estimates of the mixture model. Two datasets were used for this performance comparison: (i) Elliptical dataset which is an artificially created dataset containing three well-separated Gauss-

Table 1. Comparison results for different initialization methods for EM algorithm on two datasets

Method	Elliptical	Iris
RS+EM	-3235 ± 14.2	-198 ± 27
K-Means+EM	-3195 ± 54	-186 ± 10
SMEM	-3123 ± 54	-178.5 ± 6
TRUST-TECH-EM	-3079 ± 0.03	-173.6 ± 11

ian clusters with 900 datapoints. The data generated from a two-dimensional, three-component Gaussian mixture distribution with mean vectors at $[0\ -2]^T$, $[0\ 0]^T$, $[0\ 2]^T$ and same diagonal covariance matrix with values 2 and 0.2 along the diagonal. All the three mixtures have uniform priors. (ii) Iris dataset which contains 150 samples, 3 classes and 4 features. For this dataset, the class labels were deleted allowing it to be treated as an unsupervised learning problem.

One can see that the TRUST-TECH-EM method not only gives better results, but it also performs consistently well compared to other popular methods proposed in the literature. More details on the experiments and the algorithmic performance are presented in Reddy et al., 2008. One other problem with the standard EM algorithm is that it might sometimes converge to the boundary of the parameter space. The boundary space problem (popularly known as the singularity problem) occurs when one of the unconstrained covariance matrices approaches zero. This can be solved by using soft constraints on the covariance matrices.

FUTURE TRENDS

The EM algorithm has been successfully used in various contexts other than GMMs. Real-world applications that arise in some popular areas (such as finance) demand the use of mixture models for Poisson and various heavy-tailed distributions. The EM algorithm has been effectively used in these contexts.

In the field of data mining, EM has been very successful in solving the parameter estimation problems for probabilistic graphical models such as Hidden Markov Models, Mixtures of Factor Analyzers, Bayesian Networks etc. For example, one of the most popular EM variant used in training hidden Markov models is the Baum-Welch algorithm. Many variants of the EM algorithm are heavily used in the context of soft-clustering and hard clustering in a wide variety of real-world applications.

There is also ongoing work on effectively combining various model selection criteria and parameter estimation in the context of the EM algorithm. This topic is one of the most active areas of research in recent times. Constrained clustering is also generating much interest from different application areas. Constrained versions of the EM algorithm are also being investigated (Shental et al., 2003).

CONCLUSION

This chapter gives an overview of the EM algorithm. The theoretical properties of the EM algorithm were derived and some of the popularly used global optimization methods in the context of the EM algorithm are discussed. The details of using the EM algorithm in the context of the finite mixture models are thoroughly investigated. A comprehensive set of derivations are also provided in the context of Gaussian mixture models. Some comparative results on the performance of the EM algorithm when used along with popular global optimization methods for obtaining maximum likelihood estimates are shown. Finally, the future research trends in the EM literature are discussed.

REFERENCES

Carson, C., Belongie, S., Greenspan, H.., & Malik, J. (2002). Blobworld: Image segmentation using expectation-maximization and its application to image querying. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(8):1026–1038.

Demspter, P., Laird, N. A., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society Series B, 39(1):1–38.

Figueiredo, M., & Jain, A. K. (2002). Unsupervised learning of finite mixture models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(3):381–396.

Li, J. Q. (1999). Estimation of Mixture Models. PhD Thesis, Department of Statistics, Yale University.

McLachlan, G., & Krishnan, T. (1997). The EM Algorithm and Extensions. John Wiley and Sons, New York.

McLachlan, G., & Peel, D. (2000). Finite Mixture Models. Wiley-Interscience.

Nigam, K., McCallum, A., Thrun, S., & Mitchell, T. (2000). Text classification from labeled and unlabeled documents using EM. Machine Learning, 39(2-3):103–134.

Pernkopf, F., & Bouchaffra, D. (2005). Genetic-based EM algorithm for learning Gaussian mixture models.

IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(8):1344–1348.

Reddy, C. K. (2007). "TRUST-TECH based Methods for Optimization and Learning", PHD Thesis, Cornell University.

Reddy, C. K., Chiang, H.D., & Rajaratnam, B. (2008). "TRUST-TECH based Expectation Maximization for Learning Finite Mixture Models", IEEE Transactions on Pattern Analysis and Machine Intelligence, 30(7): 1146-1157.

Reddy, C. K., Weng, Y. C., & Chiang, H. D. (2006). Refining motifs by improving information content scores using neighborhood profile search. BMC Algorithms for Molecular Biology, 1(23):1–14.

Redner, R. A., & Walker, H. F. (1984). Mixture densities, maximum likelihood and the EM algorithm. SIAM Review, 26:195–239.

Shental, N., Bar-Hillel, A., Hertz, T., & Weinshall, D. (2003). Computing Gaussian Mixture Models with EM Using Equivalence Constraints. Neural Information Processing systems.

Ueda, N., & Nakano, R. (1998). Deterministic annealing EM algorithm. Neural Networks, 11(2):271–282.

Verbeek, J. J., Vlassis, N., & Krose, B. (2003). Efficient greedy learning of Gaussian mixture models. Neural Computation, 15(2):469–485.

Wasserman, L. (2004). All of Statistics: A concise course in statistical inference, Springer, New York.

Xu, L., & Jordan, M. I. (1996). On convergence properties of the EM algorithm for Gaussian mixtures. Neural Computation, 8(1):129–151.

Zhang, B., Zhang, C., & Yi, X. (2004). Competitive EM algorithm for finite mixture models. Pattern Recognition, 37(1):131–144.

KEY TERMS

EM Algorithm: The expectation maximization (EM) algorithm is an iterative method for finding maximum likelihood estimates.

Global Optimization: Global optimization is the task of obtaining the absolutely best set of parameters that attain the highest (or lowest) value of an objective function.

Jensen's Inequality: Jensen's inequality relates the expected value of a convex(concave) function of a random variable to the convex(concave) function of the expected value of the random variable. It emphasizes the fact that in general the expected value of a function of a random variable is not equal to the function of the expected value.

KL-Divergence: The Kullback–Leibler divergence or relative entropy is a measure of the difference between two probability distributions. It is not a metric on the space of probability distributions.

Local Maximum: A local maximum is a point which yields the highest value of a given objective function within a neighborhood region. A function has a local maximum point at x^*, if $f(x^*) \geq f(x)$ when $|x - x^*| < \varepsilon$, for some $\varepsilon > 0$. The value of the function at this point is called the local maximum of the function.

Log-Likelihood Surface: The log-likelihood surface is the log of the likelihood function which is defined as the probability density or mass function for a given data set viewed as a function of the parameters.

Maximum Likelihood: Maximum likelihood is a commonly used parameter estimation method in statistics (and allied fields) which for given data and underlying probability model chooses the parameter value under which the given data has the highest probability of occurring. The maximum likelihood estimate is not always uniquely defined.

Mixture Models: In probability and statistics, a mixture model is a probabilistic model with a distribution that is a convex combination (or weighted sum) of other distributions.

APPENDIX: PROOF OF THEOREM 1

In this appendix, we proceed to demonstrate that at every EM iteration the likelihood value does not decrease.

Theorem 1: $L(\theta_{t+1}) \geq L(\theta_t)$ for all t [or equivalently, $l(\theta_{t+1}) - l(\theta_t) \geq 0$]

We first state three simple lemmas which we shall use in the proof.

Lemma 1:

$$Q(\theta_t;\theta_t) = E_{\theta_t}\left[\log\frac{f(x,y;\theta_t)}{f(x,y;\theta_t)}\middle|X=x\right] = E_{\theta_t}[\log 1 \ \middle|X=x] = 0$$

Lemma 2: (Jensen's inequality):

Given a random variable Z and a concave function $g(.)$ defined on the support of Z, then $g(E[Z]) \geq E[g(Z)]$ (see Wasserman, 2004).

Lemma 3:

$$E_{\theta_t}\left[\log\frac{f(y|x,\theta)}{f(y|x,\theta_t)}\right] \leq 0$$

Proof of Lemma 3:

$$\begin{aligned} E_{\theta_t}\left[\log\frac{f(y|x,\theta)}{f(y|x,\theta_t)}\right] &\leq \log E_{\theta_t}\left[\frac{f(y|x,\theta)}{f(y|x,\theta_t)}\right], \quad \textit{by Jensen's inequality} \\ &= \log\left[\int \frac{f(y|x,\theta)}{f(y|x,\theta_t)} f(y|x,\theta_t)dy\right] \\ &= \log\left[\int f(y|x,\theta)dy\right] \\ &= \log 1, \ \textit{since } f(y|x,\theta) \textit{ is a density function} \\ &= 0 \end{aligned}$$

Note: The above lemma proves that the Kullback-Leibler divergence is always non-negative.

We now have all the ingredients to prove theorem 1.

Proof of Theorem 1

First note that

$$\begin{aligned} Q(\theta;\theta_t) &= E_{\theta_t}\left[\log\frac{f(x,y;\theta)}{f(x,y;\theta_t)}\middle|X=x\right] \\ &= E_{\theta_t}\left[\log\frac{f(x,\theta)f(y|x,\theta)}{f(x,\theta_t)f(y|x,\theta_t)}\middle|X=x\right] \\ &\quad \textit{by the definition of conditional distributions} \\ &= \log\left[\frac{f(x;\theta)}{f(x;\theta_t)}\right] + E_{\theta_t}\left[\log\frac{f(y|x,\theta)}{f(y|x,\theta_t)}\right] \end{aligned}$$

Therefore,

(2)

$$l(\theta) - l(\theta_t) = \log\left[\frac{f(x;\theta)}{f(x;\theta_t)}\right] = Q(\theta;\theta_t) - E_{\theta_t}\left[\log\frac{f(y|x,\theta)}{f(y|x,\theta_t)}\right]$$

Now consider the term $l(\theta_{t+1})$ - $l(\theta_t)$ in eqn(2) above with θ_{t+1} substituted for θ. First we know from lemma 3 that the expectation term on the RHS of eqn(2) is always non-positive – thus subtracting this term from $Q(\theta;\theta_t)$ amounts to adding a non-negative quantity to $Q(\theta;\theta_t)$. Note from the M-step in the EM algorithm that we obtain θ_{t+1} by maximizing $Q(\theta;\theta_t)$ w.r.t θ. Since from lemma 1, $Q(\theta_t;\theta_t) = 0$ the maximum value of $Q(\theta;\theta_t)$ w.r.t θ given by $Q(\theta_{t+1};\theta_t) \geq Q(\theta_t;\theta_t) = 0$. So both terms on the RHS of eqn(2) give rise to non-negative quantities. Hence $l(\theta_{t+1}) - l(\theta_t) \geq 0$, thus proving that at each iteration the EM step can only increase the likelihood value.

A more intuitive interpretation of the working of the EM algorithm stems from the following argument. Note from eqn(2) we can write

$$\begin{aligned} l(\theta) &= l(\theta_t) + Q(\theta;\theta_t) - E_{\theta_t}\left[\log\frac{f(y|x,\theta)}{f(y|x,\theta_t)}\right] \\ &\geq l(\theta_t) + Q(\theta;\theta_t) \quad \textit{since } E_{\theta_t}\left[\log\frac{f(y|x,\theta)}{f(y|x,\theta_t)}\right] \leq 0 \\ &\equiv g(\theta) \end{aligned}$$

Now maximizing $Q(\theta;\theta_t)$ w.r.t θ in the M-step is equivalent to maximizing $g(\theta)$ since the additional terms $l(\theta_t)$ is a constant function of θ. Examining the function $g(\theta)$ can yield insights into the workings of the EM algorithm. First note that the function $g(\theta)$ is always bounded above by the log likelihood function $l(\theta)$. Second note that at the current value $\theta = \theta_t$ the original log likelihood function $l(\theta)$ and $g(\theta)$ are equal, i.e. $l(\theta_t) = g(\theta_t)$, since $g(\theta_t) = l(\theta_t) + Q(\theta_t;\theta_t) = l(\theta_t)$ because $Q(\theta_t;\theta_t) = 0$ (see lemma 1). So at each EM cycle the two functions start off at the same value and since $l(\theta) \geq g(\theta)$, increasing $g(\theta)$ guarantees that the value of $l(\theta)$ also increases. Figure 1 in the main text gives a graphical illustration of a single EM step.

Time-Constrained Sequential Pattern Mining

Ming-Yen Lin
Feng Chia University, Taiwan

INTRODUCTION

Sequential pattern mining is one of the important issues in the research of data mining (Agrawal & Srikant, 1995; Ayres, Gehrke, & Yiu, 2002; Han, Pei, & Yan, 2004; Lin & Lee, 2004; Lin & Lee, 2005b; Roddick & Spiliopoulou, 2002). A typical example is a retail database where each record corresponds to a customer's purchasing sequence, called data sequence. A data sequence is composed of all the customer's transactions ordered by transaction time. Each transaction is represented by a set of literals indicating the set of items (called itemset) purchased in the transaction. The objective is to find all the frequent sub-sequences (called sequential patterns) in the sequence database. Whether a sub-sequence is frequent or not is determined by its frequency, named support, in the sequence database.

An example sequential pattern might be that 40% customers bought PC and printer, followed by the purchase of scanner and graphics-software, and then digital camera. Such a pattern, denoted by <(PC, printer)(scanner, graphics-software)(digital camera)>, has three elements where each element is an itemset. Although the issue is motivated by the retail industry, the mining technique is applicable to domains bearing sequence characteristics, including the analysis of Web traversal patterns, medical treatments, natural disasters, DNA sequences, and so forth.

In order to have more accurate results, constraints in addition to the support threshold need to be specified in the mining (Pei, Han, & Wang, 2007; Chen & Yu, 2006; Garofalakis, Rastogi, & Shim, 2002; Lin & Lee, 2005a; Masseglia, Poncelet, & Teisseire, 2004). Most time-independent constraints can be handled, without modifying the fundamental mining algorithm, by a post-processing on the result of sequential pattern mining without constraints. Time-constraints, however, cannot be managed by retrieving patterns because the support computation of patterns must validate the time attributes for every data sequence in the mining process. Therefore, time-constrained sequential pattern mining (Lin & Lee, 2005a; Lin, Hsueh, & Chang, 2006; Masseglia, Poncelet, & Teisseire, 2004;) is more challenging, and more important in the aspect of temporal relationship discovery, than conventional pattern mining.

BACKGROUND

The issue of mining sequential patterns with time constraints was first addressed by Srikant and Agrawal in 1996 (Srikant & Agrawal 1996). Three time constraints including minimum gap, maximum gap and sliding time-window are specified to enhance conventional sequence discovery. For example, without time constraints, one may find a pattern <(b, d, f)(a, e)>. However, the pattern could be insignificant if the time interval between (b, d, f) and (a, e) is too long. Such patterns could be filtered out if the maximum gap constraint is specified.

Analogously, one might discover the pattern <(b, c, e)(d, g)> from many data sequences consisting of itemset (d, g) occurring one day after the occurrence of itemset (b, c, e). Nonetheless, such a pattern is a false pattern in discovering weekly patterns, i.e. the minimum gap of 7 days. In other words, the sale of (b, c, e) might not trigger the sale of (d, g) in next week. Therefore, time constraints including maximum gap and minimum gap should be incorporated in the mining to reinforce the accuracy and significance of mining results.

Moreover, conventional definition of an element of a sequential pattern is too rigid for some applications. Essentially, a data sequence is defined to support a pattern if each element of the pattern is contained in an individual transaction of the data sequence. However, the user may not care whether the items in an element (of the pattern) come from a single transaction or from adjoining transactions of a data sequence if the adjoining transactions occur close in time (within a specified time interval). The specified interval is named sliding time-

window. For instance, given a sliding time-window of 6, a data sequence $< {}_{t1}(a, e)\ {}_{t2}(b)\ {}_{t3}(f)>$ can support the pattern <(a, b, e)(f)> if the difference between time t1 and time t2 is no greater than 6. Adding sliding time-window constraint to relax the definition of an element will broaden the applications of sequential patterns.

In addition to the three time constraints, duration and exact gap constraints are usually specified for finding actionable patterns (Lin, Hsueh, & Chang, 2006; Zaki, 2000). Duration specifies the maximum total time-span allowed for a pattern. A pattern having transactions conducted over one year will be filter out if the duration of 365 days is given. Exact gap can be used to find patterns, within which adjacent transactions occur exactly the specified time difference. The discovery of sequential patterns with additionally specified time constraints is referred to as mining time-constrained sequential patterns.

A typical example of mining time-constrained sequential patterns might be that finding out frequent sub-sequences having minimum support of 40%, minimum gap of 7 days, maximum gap of 30 days, sliding time-window of 2 days, and duration of 90 days.

MAIN FOCUS

Sequential pattern mining is more complex than association rule mining because the patterns are formed not only by combinations of items but also by permutations of itemsets. The number of potential sequences is by far larger than that of potential itemsets. Given 100 possible items in the database, the total number of possible itemsets is

$$\sum_{i=0}^{100}\binom{100}{i} = 2100.$$

Let the size of a sequence (sequence size) be the total number of items in that sequence. The number of potential sequences of size k is

$$\sum_{i_1=1}^{k}\binom{100}{i_1}\sum_{i_2=1}^{k-i_1}\binom{100}{i_2}\sum_{i_3=1}^{k-i_1-i_2}\binom{100}{i_3}\cdots\sum_{i_k=1}^{k-i_1-\cdots-i_{k-1}}\binom{100}{i_k}.$$

The total number of potential sequences, accumulating from size one to size 100 and more, could be enormous. Adding time constraints complicates the mining much more so that the focus of time-constrained sequential pattern mining is to design efficient algorithms for mining large sequence databases. In general, these algorithms can be categorized into Apriori based and pattern-growth based approaches, as well as vertical mining approaches.

Apriori-Based Approaches

Although there are many algorithms dealing with sequential pattern mining, few handle the mining with the addition of time constraints. The GSP (Generalized Sequential Pattern) algorithm (Srikant & Agrawal 1996) is the first algorithm that discovers sequential patterns with time constraints (including minimum gap, maximum gap, and sliding time-window) within Apriori framework. GSP solves the problem by generating and testing candidate patterns in multiple database scans and it scans the database k times to discover patterns having k items. Candidate patterns having any non-frequent sub-sequence are pruned before testing to reduce the search space. In a database scan, each data sequence is transformed into items' transaction-time lists for fast finding of certain element with a time tag. Since the start-time and end-time of an element (may comprise several transactions) must be considered, GSP defines 'contiguous sub-sequence' for candidate generation, and move between 'forward phase' and 'backward phase' for checking whether a data sequence contains a certain candidate.

Pattern-Growth Based Approaches

A general pattern-growth framework was presented for constraint-based sequential pattern mining (Pei, Han, & Wang, 2007). From the application point of view, seven categories of constrains including item, length, super-pattern, aggregate, regular expression, duration, and gap constraints were covered. Among these constraints, duration and gap constraints are tightly coupled with the support counting process because they confine how a data sequence contains a pattern. Orthogonally classifying constraints by their roles in mining, monotonic, anti-monotonic, and succinct constraints were characterized and the prefix-monotone constraint was introduced. The prefix-growth framework which pushes prefix-monotone constraints into PrefixSpan (Pei, Han, Mortazavi-Asl, Wang, Pinto,

Chen, Dayal, & Hsu, 2004) was also proposed in (Pei, Han, & Wang, 2007). However, with respect to time constraints, prefix-growth mentioned only minimum gap and maximum gap time constraints (though duration constraint was addressed). The sliding time-window was not considered at all since the constraint is neither monotonic nor anti-monotonic.

The DELISP (DELImited Sequential Pattern mining) algorithm (Lin & Lee, 2005a), fully functionally equivalent to GSP on time constraint issues, solves the problem within the pattern-growth framework. DELISP solves the problem by recursively growing valid patterns in projected sub-databases generated by sub-sequence projection. To accelerate mining by reducing the size of sub-sequences, the constraints are integrated in the projection to delimit the counting and growing of sequences. The main idea of DELISP is efficiently 'finding' the frequent items, and then effectively 'growing' potential patterns in the sub-databases constructed by projecting sub-sequences corresponding to the frequent items. The time-tags are also projected into the sub-databases to generate patterns satisfying the time constraints. DELISP decomposes the mining problem by recursively growing patterns, one item longer than the current patterns, in the projected sub-databases. However, the potential items used to grow are subjected to minimum gap and maximum gap constraints, called de-limited growth. On projecting sub-databases, bi-directional growth is avoided by imposing the item-order in the growth. Only items having order lexicographically larger than the order of the existing items need to be projected to grow the itemset-typed patterns. Such a projection is referred to as windowed-projection. In DELISP, the bounded projection technique eliminates invalid sub-sequence projections caused by unqualified maximum/minimum gaps, the windowed projection technique reduces redundant projections for adjacent elements satisfying the sliding time-window constraint, and the delimited growth technique grows only the patterns satisfying constraints.

The METISP (MEmory Time-Indexing for Sequential Pattern mining) algorithm (Lin, Hsueh, & Chang, 2006) also finds out time-constrained sequential patterns. METISP uses the pattern-growth strategy similar to PrefixSpan and DELISP algorithms. METISP algorithm solves time constraints including minimum/maximum/exact gap, sliding time-window and duration constraints at the same time. The technique of memory indexing is used to mark the timestamps (called time-index) for fast growth of patterns in memory. Hence, the inherent properties of time constraints can be applied to reduce the search space and improve the efficiency of mining. METISP first loads DB into memory and scans the in-memory database once to find all frequent items. With respect to each frequent item, METISP then constructs a time index-set for the sequences of single items and recursively forms time-constrained sequential patterns of longer length. The time index-set is a set of (data-sequence pointer, time index) pairs. Only those data sequences containing that item would be included. The time index indicates the list of *it:lst:let* triplets, where *it* represents the initial time, *lst* the last start-time, and *let* the last end-time of a pattern. For extra-large database, the algorithm applies a partition-and-verification technique for mining databases that cannot fit into the memory. The extra-large database is partitioned so that each partition can be handled, in memory, by the METISP algorithm. The potential time-constrained sequential patterns in each partition are collected and served as candidates to be validated in the second pass of the entire database scanning. The memory time index sets are effective to handle these time constraints within the pattern-growth framework without generating any candidates or sub-databases in disk.

Vertical Mining Approaches

The cSPADE (constrained SPADE) algorithm (Zaki 2001) extends the vertical mining algorithm SPADE (Sequential PAttern Discovery using Equivalence classes) (Zaki 2000) to deal with time constraints. Vertical mining approaches discover sequential patterns using join-operations and vertical database layout, where data sequences are transformed into items' (sequence-id, time-id) lists. The cSPADE algorithm checks minimum and maximum gaps while doing temporal joins. The huge sets of frequent 2-sequences must be preserved to generate the required classes for the maximum gap constraint. However, it does not appear feasible for cSPADE to incorporate the sliding time-window by expanding the id-lists and augmenting the join-operations with temporal information. The sliding time-window constraint was not mentioned in cSPADE.

FUTURE TRENDS

Time-constrained sequential pattern mining will become one of the essential components of any sequential pattern mining mechanism. More constraints, theories, and techniques of time-constrained sequential pattern mining will be developed in the near future, followed by comprehensive comparisons of these theories and techniques. Efficient and effective mining algorithms will also be developed not only for conventional databases but also for data stream applications (Jiang & Gruenwald, 2006). Queries involving time constraints will also be expanded for practical systems.

CONCLUSION

In order to provide users with more flexibility and more accurate results on the discovery of sequential patterns, time constraints should be incorporated into the mining. Given that time-constrained sequential pattern mining is more complicated than traditional sequential pattern mining, some algorithms were developed. In general, mining sequence databases could be solved under the Apriori or pattern-growth framework. Apriori based approaches generate candidates and have to check the time constraints for every data sequence. Pattern-growth based approaches recursively mine patterns either in the disk-projected databases or in memory using the memory index. Pattern-growth based algorithms generally perform better than Apriori ones. Still, more efforts are desirable to solve the complicated mining problem, in conventional static databases or in the evolving data streams.

REFERENCES

Agrawal R. & Srikant R. (1995) Mining sequential patterns. In: Yu P. S. & Chen A.L.P. (Eds.), *Proceedings of the 11th international conference on data engineering*, 3-14.

Ayres J., Gehrke J., Yiu T., & Flannick J. (2002) Sequential pattern mining using bitmaps. In: *Proceedings of the 8th ACM SIGKDD international conference on knowledge discovery and data mining*.

Chen Y. L. & Hu Y.H. (2006). Constraint-based sequential pattern mining: The consideration of recency and compactness. *Decision Support Systems, 42*(2), 1203-1215.

Garofalakis M.N., Rastogi R., & Shim K. (2002) Mining sequential patterns with regular expression constraints. *IEEE Transactions on Data and Knowledge Engineering, 14*(3), 530-552.

Han J., Pei J. & Yan X.(2004) From sequential pattern mining to structured pattern mining: A pattern-growth approach. *Journal of Computer Science and Technology, 19*(3), 257-279.

Jiang N. & Gruenwald L. (2006) Research issues in data stream association rule mining. *SIGMOD Record, 35*(1), 14-19.

Lin M. Y. & Lee S. Y. (2004) Incremental update on sequential patterns in large databases by implicit merging and efficient counting. *Information Systems, 29*(5), 385-404.

Lin M.Y., & Lee S.Y. (2005a) Efficient mining of sequential patterns with time constraints by delimited pattern growth. *Knowledge and Information Systems, 7*(4), 499-514.

Lin M.Y. & Lee S.Y. (2005b) Fast discovery of sequential patterns by memory indexing and database partitioning. *Journal of Information Science and Engineering, 21*(1), 109-128.

Lin M. Y., Hsueh S. C., & Chang C. W. (2006) Fast discovery of time-constrained sequential patterns using time-indexes. In: Li X., Zaïane O. R., & Li Z. (Eds.) *Lecture Notes in Computer Science 4093*. 693-701.

Masseglia F., Poncelet P., & Teisseire M. (2004) Pre-processing time constraints for efficiently mining generalized sequential patterns. In: *Proceedings of 11th international symposium on temporal representation and reasoning*. 87-95.

Pei J., Han J., Mortazavi-Asl B., Wang J., Pinto H., Chen Q., Dayal U., & Hsu M-C. (2004) Mining sequential patterns by pattern-growth: The PrefixSpan approach. *IEEE Transactions on Knowledge and Data Engineering, 16*(11), 1424-1440.

Pei J., Han J., Wang W. (2007) Constraint-based sequential pattern mining: the pattern-growth methods. *Journal of Intelligent Information Systems, 28*(2), 133-160.

Roddick J.F., & Spiliopoulou M. (2002) A survey of temporal knowledge discovery paradigms and methods. *IEEE Transactions on Knowledge and Data Engineering*, *14*(4), 750-768.

Srikant R. & Agrawal R. (1996) Mining sequential patterns: Generalizations and performance improvements. In: *Proceedings of the 5th international conference on extending database technology*. 3-17.

Zaki M.J. (2001) SPADE: An efficient algorithm for mining frequent sequences. *Machine Learning Journal*, *42*(1), 31-60.

Zaki M.Z. (2000) Sequence mining in categorical domains: Incorporating constraints. In *Proceedings of the 9th international conference on information and knowledge management*. 422-429.

KEY TERMS

Duration: The maximum allowed time difference between the first and the latest occurrences of elements in the entire sequence.

Exact Gap: The time difference between the occurrences of any two adjacent elements.

Maximum Gap: The maximum allowed time difference between the earliest occurrence of an item in an element and the latest occurrence of an item in the immediately succeeding element.

Minimum Gap: The minimum allowed time difference between the latest occurrence of an item in an element and the earliest occurrence of an item in the immediately succeeding element.

Sequential Pattern: A sequence whose occurrence frequency in a sequence database is no less than a user-specified frequency threshold.

Sliding Time-Window: The maximum allowed time difference between the first and the latest occurrences of elements to be considered as a single element.

Time-Constrained Sequential Pattern: The sequential pattern that additionally satisfies the user-specified time constraints.

Time-Index: An index that indicates the occurrences of a pattern for a data sequence.

Topic Maps Generation by Text Mining

Hsin-Chang Yang
Chang Jung University, Taiwan

Chung-Hong Lee
National Kaohsiung University of Applied Sciences, Taiwan

INTRODUCTION

Topic maps provide a general, powerful, and user-oriented way to navigate the information resources under consideration in any specific domain. A topic map provides a uniform framework that not only identifies important subjects from an entity of information resources and specifies the resources that are semantically related to a subject, but also explores the relations among these subjects. When a user needs to find some specific information on a pool of information resources, he or she only needs to examine the topic maps of this pool, select the topic that seems interesting, and the topic maps will display the information resources that are related to this topic, as well as its related topics. The user will also recognize the relationships among these topics and the roles they play in such relationships. With the help of the topic maps, you no longer have to browse through a set of hyperlinked documents and hope that you may eventually reach the information you need in a finite amount of time, while knowing nothing about where to start. You also don't have to gather some words and hope that they may perfectly symbolize the idea you're interested in, and be well-conceived by a search engine to obtain reasonable result. Topic maps provide a way to navigate and organize information, as well as create and maintain knowledge in an infoglut.

To construct a topic map for a set of information resources, human intervention is unavoidable at the present time. Human effort is needed in tasks such as selecting topics, identifying their occurrences, and revealing their associations. Such a need is acceptable only when the topic maps are used merely for navigation purposes and when the volume of the information resource is considerably small. However, a topic map should not only be a topic navigation map. The volume of the information resource under consideration is generally large enough to prevent the manual construction of topic maps. To expand the applicability of topic maps, some kind of automatic process should be involved during the construction of the maps. The degree of automation in such a construction process may vary for different users with different needs. One person may need only a friendly interface to automate the topic map authoring process, while another may try to automatically identify every component of a topic map for a set of information resources from the ground up. In this article, we recognize the importance of topic maps not only as a navigation tool but also as a desirable scheme for knowledge acquisition and representation. According to such recognition, we try to develop a scheme based on a proposed text-mining approach to automatically construct topic maps for a set of information resources. Our approach is the opposite of the navigation task performed by a topic map to obtain information. We extract knowledge from a corpus of documents to construct a topic map. Although currently the proposed approach cannot fully construct the topic maps automatically, our approach still seems promising in developing a fully automatic scheme for topic map construction.

BACKGROUND

Topic map standard (ISO, 2000) is an emerging standard, so few works are available about the subject. Most of the early works about topic maps focus on providing introductory materials (Ahmed, 2002; Pepper, 1999; Beird, 2000; Park & Hunting, 2002). Few of them are devoted to the automatic construction of topic maps. Two works that address this issue were reported in Rath (1999) and Moore (2000). Rath discussed a framework for automatic generation of topic maps according to a so-called topic map template and a set of generation rules. The structural information of topics is maintained in the template. To create the topic map, they used a generator to interpret the generation rules and

extract necessary information that fulfills the template. However, both the rules and the template are to be constructed explicitly and probably manually. Moore discussed topic map authoring and how software may support it. He argued that the automatic generation of topic maps is a useful first step in the construction of a production topic map. However, the real value of such a map comes through the involvement of people in the process. This argument is true if the knowledge that contained in the topic maps can only be obtained by human efforts. A fully automatic generation process is possible only when such knowledge may be discovered from the underlying set of information resources through an automated process, which is generally known as knowledge discovery from texts, or *text mining* (Hearst, 1999; Lee & Yang, 1999; Wang, 2003; Yang & Lee, 2000).

MAIN THRUST

We briefly describe the text-mining process and the generation process of topic maps in this section.

The Text-Mining Process

Before we can create topic maps, we first perform a text-mining process on the set of information resources to reveal the relationships among the information resources. Here, we only consider those information resources that can be represented in regular texts. Examples of such resources are Web pages, ordinary books, technical specifications, manuals, and so forth. The set of information resources is collectively known as *the corpus*, and individual resource is referred to as a *document* in the following text. To reveal the relationships between documents, the popular *self-organizing map (SOM)* algorithm (Kohonen, Kaski, Lagus, Salojärvi, Honkela, Paatero, & Saarela, 2000) is applied to the corpus to cluster documents. We adopt the vector space model (Baeza-Yates and Ribiero-Neto, 1999) to transform each document in the corpus into a binary vector. These document vectors are used as input to train the map. We then apply two kinds of labeling processes to the trained map and obtain two feature maps, namely the *document cluster map (DCM)* and the *word cluster map (WCM)*. In the document cluster map, each neuron represents a document cluster that contains several similar documents with high word co-occurrence. In the word cluster map, each neuron represents a cluster of words revealing the general concept of the corresponding document cluster that is associated with the same neuron in the document cluster map.

The text-mining process described in the preceding paragraph provides a way for us to reveal the relationships between the topics of the documents. Here, we introduce a method to identify topics and the relationships between them. The method also arranges these topics in a hierarchical manner according to their relationships. As we mention earlier in this article, a neuron in the DCM represents a cluster of documents containing words that often co-occurred in these documents. Besides, documents that associate with neighboring neurons contain similar sets of words. Thus, we may construct a supercluster by combining neighboring neurons. To form a supercluster, we first define the distance between two clusters:

$$D(i,j) = H\left(\left\|\mathbf{G}_i - \mathbf{G}_j\right\|\right), \tag{1}$$

where i and j are the neuron indices of the two clusters, and $\mathbf{G}_i$ is the two-dimensional grid location of neuron i. $\|\mathbf{G}_i - \mathbf{G}_j\|$ measures the Euclidean distance between the two coordinates $\mathbf{G}_i$ and $\mathbf{G}_j$. $H(x)$ is a bell-shaped function that has a maximum value when x=0. We also define the dissimilarity between two clusters:

$$\delta(i,j) = \left\|\mathbf{w}_i - \mathbf{w}_j\right\|, \tag{2}$$

where $\mathbf{w}_i$ denotes the synaptic weight vector of neuron i. We may then compute the *supporting cluster similarity*, S_i, for a neuron i from its neighboring neurons by the equations

$$s(i,j) = \frac{\mathrm{doc}(i)\mathrm{doc}(j)}{F\left(D(i,j)\delta(i,j)\right)}$$
$$S_i = \sum_{j \in B_i} s(i,j) \tag{3}$$

where doc(i) is the number of documents associated with neuron i in the document cluster map, and B_i is the set of neuron indices in the neighborhood of neuron i. The function F: $\mathbf{R}^+ \rightarrow \mathbf{R}^+$ is a monotonically increasing function. A *dominating neuron* is the neuron that has locally maximal supporting cluster similarity.

We may select dominating neurons by the following algorithm:

- **Step 1:** Find the neuron with the largest supporting cluster similarity. Select this neuron as the dominating neuron.
- **Step 2:** Eliminate its neighbor neurons so they will not be considered as dominating neurons.
- **Step 3:** If no neuron is left, or the number of dominating neurons exceeds a predetermined value, stop. Otherwise, go to Step 1.

A dominating neuron may be considered as the centroid of a supercluster, which contains several clusters. We assign every cluster to some supercluster by the following method. The *i*th cluster (neuron) is assigned to the *k*th supercluster if:

$$\delta(i,k) = \min_{l} \delta(i,l),\ l \text{ is a super-cluster.} \quad (4)$$

A supercluster may be thought of as a category that contains several subcategories. Let C_k denote the set of neurons that belong to the *k*th supercluster, or category. The category topics are selected from those words that associate with these neurons in the WCM. For all neurons $j \in C_k$, we select the n^*th word as the category topic if:

$$\sum_{j \in C_k} w_{j_{n^*}} = \max_{1 \le n \le N} \sum_{j \in C_k} w_{j_n} \quad (5)$$

Equation 5 selects the word that is the most important to a supercluster, because the components of the synaptic weight vector of a neuron reflect the willingness that the neuron wants to learn the corresponding input data, that is, words.

The topics that are selected by Equation 5 form the top layer of the category hierarchy. To find the descendants of these topics in the hierarchy, we may apply the above process to each supercluster and obtain a set of subcategories. These subcategories form the new superclusters that are on the second layer of the hierarchy. The category structure can then be revealed by recursively applying the same category generation process to each newfound supercluster. We decrease the size of the neighborhood in selecting dominating neurons when we try to find the subcategories.

Automatic Topic Maps Construction

The text-mining process described in the preceding section reveals the relationships between documents and words. Furthermore, it may identify the topics in a set of documents, reveals the relationships among the topics, and arranges the topics in a hierarchical manner. The result of such a text-mining process can be used to construct topic maps. We discuss the steps in topic map construction in the following subsections.

Identifying Topics and Topic Types

The topics in the constructed topic map can be selected as the topics identified by Equation 5. All the identified topics in every layer of the hierarchy can be used as topics. Because topics in different layers of the hierarchy represent different levels of significance, we may constrain the significance of topics in the map by limiting the depth of hierarchy from which we select topics. If we only used topics in higher layers, the number of topics is small, but those topics represent more important topics. The significance level can be set explicitly in the beginning of the construction process or determined dynamically during the construction process. One way to determine the number of topics is by considering the size of the self-organizing map.

The topic types can also be determined by the constructed hierarchy. As we mention earlier in this paper, a topic on higher layers of the hierarchy represents a more important concept than those on lower layers. For a parent-child relationship between two concepts on two adjacent layers, the parent topic should represent an important concept of its child topic. Therefore, we may use the parent topic as the type of its child topics. Such usage also fulfills the requirement of the topic map standard (that a topic type is also a topic).

Identifying Topic Occurrences

The occurrences of an identified topic are easy to obtain after the text-mining process. Because a topic is a word labeled to a neuron in the WCM, its occurrences can be assigned as the documents labeled to the same neuron in the DCM. That is, let a topic t be labeled to neuron *A* in the WCM, and the occurrences of t should be those documents labeled to the same neuron *A* in the DCM. For example, if the topic 'text mining' was labeled to the 20th neuron in the WCM, all the docu-

ments labeled to the 20th neuron in the DCM should be the occurrences of this topic. Furthermore, we may create more occurrences of this topic by allowing the documents labeled to lower levels of the hierarchy to also be included. For example, if neuron 20 in the preceding example were located on the second level of a topic hierarchy, we could also allow the clusters of documents associated with topics below this level to be occurrences of this topic. Another approach is to use the DCM directly, such that we also include the documents associated with the neighboring neurons as its occurrences.

Identifying Topic Associations

The associations among topics can be identified in two ways with our method. The first is to use the developed hierarchy structure among topics. A topic is associated with the other if a path exists between them. We should limit the lengths of such paths to avoid establishing associations between pairs of unrelated topics. For example, if we limited the length to 1, only topics that are direct parents and children are associated with the topic under consideration. The type of such associations is essentially an instance-class association. The second way to identify topic associations simply examines the WCM and finds the associations. To establish associations to a topic t, we first find the neuron *A* to which t is labeled. We then establish associations between t and every topic associated with some neighboring neuron of *A*. The neighboring neurons are selected from a neighborhood of *A* that is arbitrarily set by the creator. Obviously, a large neighborhood will create many associations. We should at least create associations between t and other topics associated with the same neuron *A*, because they are considered closely related topics in the text-mining process. The association types are not easy to reveal by this method, because we do not fully reveal the semantic relations among neurons after the text-mining process. An alternative method to determine the association type between two topics is to use the semantic relation defined in a well-developed ontology, such as WordNet (Fellbaum, 1998).

FUTURE TRENDS

Topic maps will be an emergent standard for information navigation in the near future. Its topic-driven navigation scheme allows the users to retrieve their documents without tedious browsing of the whole infoglut. However, the generation of topic maps still dominates their spread. An editor will help, provided it can generate the necessary ingradients of a topic map automatically, or at least semiautomatically. However, such a generation process is difficult, because we need to reveal the semantics of the documents. In this aspect, the data-mining techniques will help. Therefore, the future trends of topic map generation should be as follows:

- Applying knowledge discovery techniques to discover topics and their associations without the intervention of human beings
- Incorporating topic map standard and Web representation languages, such as XML, to promote the usage of topic maps
- Developing a user interface that allows users to create and edit the topic maps with the aid of automatic generated ingredients
- Developing a tool to integrate or migrate existing topic maps
- Constructing metadata in topic maps for applications on the semantic Web (Daconta, Obrst, & Smith, 2003)
- Mining the existing topic maps from their structures and ingredients

CONCLUSION

In this paper, we present a novel approach for semiautomatic topic map construction. The approach starts from applying a text-mining process to a set of information resources. Two feature maps, namely the document cluster map and the word cluster map, are created after the text-mining process. We then apply a category hierarchy development process to reveal the hierarchical structure of the document clusters. Some topics are also identified by such a process to indicate the general subjects of those clusters located in the hierarchy. We may then automatically create topic maps according to the two maps and the developed hierarchy. Although our method may not identify all the kinds of components that should construct a topic map, our approach seems promising because the text-mining process achieves satisfactory results in revealing implicit topics and their relationships.

REFERENCES

Ahmed, K. (2002). Introducing topic maps. *XML Journal 3*(10), 22-27.

Baeza-Yates, R., & Ribiero-Neto, B. (1999). In *Modern information retrieval* (Chapter 2). Reading, MA: Addison-Wesley.

Beird, C. (2000). Topic map cartography. *Proceedings of the XML Europe 2000 GCA Conference*, Paris, June 12-16.

Daconta, M. C., Obrst, L. J., & Smith, K. T. (2003). *The semantic Web: A guide to the future of XML, Web services, and knowledge management.* Indianapolis: Wiley.

Fellbaum, C. (1998). *WordNet: An electronic lexical database.* Cambridge, MA: MIT Press.

Hearst, M. A. (1999). Untangling text data mining. *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, College Park, Maryland, USA, June 20-26.

ISO (2000). ISO/IEC 13250, *Information technology - SGML Applications - Topic Maps.* Geneva, Switzerland: ISO.

Kohonen, T., Kaski, S., Lagus, K., Salojärvi, J., Honkela, J., Paatero, V., & Saarela, A. (2000). Self organization of a massive document collection. *IEEE Transactions on Neural Networks, 11*(3), 574-585.

Lee, C. H., & Yang, H. C. (1999). A web text mining approach based on a self-organizing map. *Proceedings of the Second ACM Workshop on Web Information and Data Management* (pp. 59-62), Kansas City, Missouri, USA, November 5-6.

Moore, G. (2000). Topic map technology — the state of the art. In *XML Europe 2000*, Paris, France.

Park, J., & Hunting, S. (2002). *XML topic maps: Creating and using topic maps for the Web*, June 12-16. Reading, MA: Addison-Wesley.

Pepper, S. (1999). Navigating haystacks, discovering needles. *Markup Languages: Theory and Practice, 1*(4), 41-68.

Rath, H. H. (1999). Technical issues on topic maps. *Proceedings of the Metastructures 1999 Conference, GCA*, Montreal, Canada, August 16-18.

Wang, J. (2003). *Data mining: Opportunities and challenges*. Hershey, PA: Idea Group.

Yang, H. C., & Lee, C. H. (2000). Automatic category structure generation and categorization of Chinese text documents. *Proceedings of the Fourth European Conference on Principles and Practice of Knowledge Discovery in Databases* (pp. 673-678), France, September 13-16.

KEY TERMS

Neural Networks: Learning systems, designed by analogy with a simplified model of the neural connections in the brain, that can be trained to find nonlinear relationships in data.

Self-Organizing Maps: A neural network model developed by Teuvo Kohonen that has been recognized as one of the most successful models. The model uses an unsupervised learning process to cluster high-dimensional data and map them into a one- or two-dimensional feature map. The relationships among data can be reflected by the geometrical distance between their mapped neurons.

Text Mining: The application of analytical methods and tools to usually unstructured textual data for the purpose of identifying patterns and relationships such as classification, prediction, estimation, or affinity grouping.

Topic Associations: The relationships between two or more topics in a topic map.

Topic Maps: A navigation scheme for exploring information resources in a topic-driven manner. When a set of information resources are provided, their topics as well as the associations among topics are identified and are used to form a map that guides the user through the topics.

Topic Occurrences: A topic may be linked to one or more information resources that are deemed to be relevant to the topic in some way. Such resources are called occurrences of the topic.

Topics: The object or node in the topic map that represents the subject being referred to. However, the relationship between topics and subjects is (or should be) one to one, with every topic representing a single subject, and every subject being represented by just one topic.

Topic Types: Topics can be categorized according to their kind. In a topic map, any given topic is an instance of zero or more topic types. This corresponds to the categorization inherent in the use of multiple indexes in a book (index of names, index of works, index of places, etc.), and to the use of typographic and other conventions to distinguish different types of topics.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 1130-1134, copyright 2005 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

Transferable Belief Model

Philippe Smets
Université Libre de Bruxelle, Belgium

INTRODUCTION

This note is a very short presentation of the transferable belief model (TBM), a model for the representation of quantified beliefs based on belief functions. Details must be found in the recent literature.

The TBM covers the same domain as the subjective probabilities except probability functions are replaced by belief functions which are much more general. The model is much more flexible than the Bayesian one and allows the representation of states of beliefs not adequately represented with probability functions. The theory of belief functions is often called the Dempster-Shafer's theory, but this term is unfortunately confusing.

The Various Dempster-Shafer's Theories

Dempster-Shafer's theory covers several models that use belief functions. Usually their aim is in the modeling of someone's degrees of belief, where a degree of belief is understood as strength of opinion. They do not cover the problems of vagueness and ambiguity for which fuzzy sets theory and possibility theory are more appropriate.

Beliefs result from uncertainty. Uncertainty can result from a random process (the objective probability case), or from a lack of information (the subjective case). These two forms of uncertainty are usually quantified by probability functions.

Dempster-Shafer's theory is an ambiguous term as it covers several models. One of them, the "transferable belief model" is a model for the representation of quantified beliefs developed independently of any underlying probability model. Based on Shafer's initial work (Shafer, 1976) it has been largely extended since (Smets,1998; Smets & Kennes, 1994; Smets & Kruse, 1997).

The Representation of Quantified Beliefs

Suppose a finite set of worlds Ω called the frame of discernment. The term "world" covers concepts like state of affairs, state of nature, situation, context, value of a variable... One world corresponds to the actual world. An agent, denoted You (but it might be a sensor, a robot, a piece of software), does not know which world corresponds to the actual world because the available data are imperfect. Nevertheless, You have some idea, some opinion, about which world might be the actual one. So for every subset A of Ω, You can express Your beliefs, i.e., the strength of Your opinion that the actual world belongs to A. This strength is denoted bel(A). The larger bel(A), the stronger You believe that the actual world belongs to A.

Credal vs. Pignistic Levels

Intrinsically beliefs are not directly observable properties. Once a decision must be made, their impact can be observed.

In the TBM, we have described a two level mental model in order to distinguish between two aspects of beliefs, belief as weighted opinions, and belief for decision making (Smets, 2002a). The two levels are: the credal level, where beliefs are held, and the pignistic level, where beliefs are used to make decisions (credal and pignistic derive from the Latin words "credo", I believe and "pignus", a wage, a bet).

Usually these two levels are not distinguished and probability functions are used to quantify beliefs at both levels. Once these two levels are distinguished, as done in the TBM, the classical arguments used to justify the use of probability functions do not apply anymore at the credal level, where beliefs will be represented by belief functions. At the pignistic level, the probability function needed to compute expected utilities are called pignistic probabilities to enhance they do not represent beliefs, but are just induced by them.

BACKGROUND

Belief Function Inequalities

The TBM is a model developed to represent quantified beliefs. The TBM departs from the Bayesian approach

in that we do not assume that bel satisfies the additivity encountered in probability theory. We get inequalities like : bel(A∪B) ≥bel(A) + bel(B) - bel(A∩B).

Basic Belief Assignment

Definition 2.2

Let Ω be a frame of discernment. A basic belief assignment (bba) is a function $m : 2^{\Omega} \rightarrow [0, 1]$ that satisfies $\Sigma_{A \subseteq \Omega} m(A) = 1$.

The term m(A) is called the basic belief mass (bbm) given to A. The bbm m(A) represents that part of Your belief that supports A, i.e., the fact that the actual world belongs to A, without supporting any more specific subset, by lack of adequate information.

As an example, consider that You learn that the actual world belongs to A, and You know nothing else about its value. Then some part of Your beliefs will be given to A, but no subset of A will get any positive support. In that case, You would have m(A) > 0 and m(B) = 0 for all B≠A, B≠Ω, and m(Ω) = 1-m(A).

Belief Functions

The bba m does not in itself quantify your belief that the actual world belongs to A. Indeed, the bbm m(B) given to any non empty subset B of A also supports that the actual world belongs to A. Hence, the degree of belief bel(A) is obtained by summing all the bbms m(B) for all B non empty subset of A. The degree of belief bel(A) quantifies the total amount of justified specific support given to the fact that the actual world belongs to A. We say justified because we include in bel(A) only the bbms given to subsets of A. For instance, consider two distinct elements x and y of Ω. The bbm m({x, y}) given to {x, y} could support x if further information indicates this. However given the available information the bbm can only be given to {x, y}. We say specific because the bbm m(Ø) is not included in bel(A) as it is given to the subsets that supports not only A but also not A.

The originality of the TBM comes from the non-null masses that may be given to non-singletons of ‡. In the special case where only singletons get positive bbms, the function bel is a probability function. In that last case, the TBM reduces itself to the Bayesian theory.

Shafer assumed m(Ø) = 0. In the TBM, such a requirement is not assumed. That mass m(Ø) reflects both the non-exhaustivity of the frame and the existence of some conflict between the beliefs produced by the various belief sources.

Expressiveness of the TBM

The advantage of the TBM over the classical Bayesian approach resides in its large flexibility, its ability to represent every state of partial beliefs, up to the state of total ignorance. In the TBM, total ignorance is represented by the vacuous belief function, i.e., a belief function such that m(Ω) = 1, m(A) = 0 for all A with A≠Ω. Hence bel(Ω) = 1 and bel(A) = 0 for all A strict subset of Ω. It expresses that all You know is that the actual world belongs to Ω. The representation of total ignorance in probability theory is hard to achieve adequately, most proposed solutions being doomed to contradictions. With the TBM, we can of course represent every state of belief, full ignorance, partial ignorance, probabilistic beliefs, or even certainty (m(A) = 1 corresponds to A is certain).

Example

Let us consider a somehow reliable witness in a murder case who testifies to You that the killer is a male. Let 0.7 be the reliability You give to the testimony (0.7 is the probability, the belief that the witness is reliable). Suppose furthermore that a priori You have an equal belief that the killer is a male or a female.

A classical probability analysis would compute the probability P(M) of M= “the killer is a male” given the witness testimony as: P(M) = P(M|Reliable)P(Reliable) + P(M|Not Reliable)P(Not Reliable)= 1.0x0.7 + 0.5x0.3 = 0.85, where “Reliable and Not Reliable refer to the witness” reliability. The value 0.85 is the sum of the probability of M given the witness is reliable (1.) weighted by the probability that the witness is reliable (0.7) plus the probability of M given the witness is not reliable (0.5, the proportion of males among the killers) weighted by the probability that the witness is not reliable (0.3).

The TBM analysis is different. You have some reason to believe that the killer is a male, as so said the witness. But this belief is not total (maximal) as the witness might be wrong. The 0.7 is the belief You give to the fact that the witness tells the truth (is reliable), in which case the killer is male. The remaining 0.3 mass is given to the fact that the witness is not really telling

the truth (he lies or he might have seen a male, but this was not the killer). In that last case, the testimony does not tell You anything about the killer's sex. So the TBM analysis will give a belief 0.7 to M: bel(M) = 0.7 (and bel(Not M) = 0).The information relative the population of killers (the 0.5) is not relevant to Your problem. Similarly, the fact that almost all crimes are committed by the members of some particular group of individuals may not be used to prove your case.

Conditioning

Suppose You have some belief on Ω represented by the bba m. Then some further evidence becomes available to You and this piece of information implies that the actual world cannot be one of the worlds in not-A. Then the mass m(B) that initially was supporting that the actual world is in B now supports that the actual world is in B∩A as every world in not-A must be "eliminated". So m(B) is transferred to B∩A after conditioning on A. (The model gets its name from this transfer operation.)

This operation leads to the conditional bba. This rule is called the Dempster's rule of conditioning.

Example

Continuing with the murder case, suppose there are only two potential male suspects: Phil and Tom, so m({Phil, Tom}) = 0.7. Then You learn that Phil is not the killer. The initial testimony now supports that the killer is Tom. The reliability 0.7 You gave to the testimony initially supported "the killer is Phil or Tom". The new information about Phil implies that the value 0.7 now supports "the killer is Tom".

After conditioning, a mass can be given to Ø. It represents the conflict between the previous beliefs given to not-A with the new conditioning piece of evidence that states that A holds. In probability theory and in the model initially developed by Shafer, this conflict is hidden. In the TBM, we keep it and use it to develop expert systems built for conflict resolutions. Note that some positive mass given to Ø may also result from the non exhaustivity of the frame of discernment.

Further Results

Since Shafer seminal work, many new concepts have been developed. For lack of space, we cannot present them. Reader is referred to the author web site for downloadable papers (http://iridia.ulb.ac.be/ ~ psmets/).

On the theoretical side, the next issues have been solved, among which:

1. The concept of open and close world assumptions, so non-exhaustive frames of discernment are allowed.
2. The disjunctive rule of combination, the general combination rules, the belief function negation.
3. The generalized Bayesian theorem to build a belief on space Y from the conditional belief on space X given each value of Y and an observation on X (Delmotte & Smets, 2004).
4. The pignistic transformation to build the probability function needed for decision-making.
5. The discounting of beliefs produced by partially reliable sources.
6. The manipulation of the conflict (Lefevre, Colot, & Vannooremberghe, 2002).
7. The canonical decompositions of any belief functions in simple support functions.
8. The specialization, cautious combinations, alpha-junctions.
9. The belief functions defined on the reals.
10. Belief ordering and least commitment principle.
11. Doxastic independence that translates stochastic independence into belief function domain (Ben Yaghlane, Smets, & Mellouli, 2001).
12. Evidential networks, directed or undirected, for the efficient propagation of beliefs in networks (Shenoy, 1997).
13. Fast Mobius transforms to transform masses into belief and plausibility functions and vice versa.
14. Approximation methods (Denoeux & Ben Yaghlane, 2002; Haenni & Lehmann, 2002).
15. Matrix notation for manipulating belief functions (Smets, 2002b).
16. Axiomatic justifications of most concepts.

The TBM has been applied to many problems among which:

1. Kalman filters and joint tracking and classifications.
2. Data association and determination of the number of detected objects (Ayoun & Smets, 2001).
3. Data clustering (Denoeux & Masson, 2004).

4. Expert systems for conflict management (Milisavljevic, Bloch, & Acheroy, 2000).
5. Belief assessment (similarity measures, frequencies).
6. TBM classifiers: case base and model base.
7. Belief decision trees (Elouedi, Mellouli, & Smets, 2001).
8. Planning and pre-posterior analyses.
9. Sensors with limited knowledge, limited communication bandwidth, self-repeating, decaying memory, varying domain granularity.
10. Tuning reliability coefficients for partially reliable sensors (Elouedi, Mellouli, & Smets, 2004).

The author has developed a computer program TBMLAB, which is a demonstrator for the TBM written in MATLAB. It is downloadable from the web site: http://iridia.ulb.ac.be/~psmets/. Many tools, tutorials and applications dealing with the TBM can be found in this freeware.

FUTURE TRENDS

The TBM is supposed to cover the same domain as probability theory, hence the task is enormous. Many problems have not yet been considered and are open for future work.

One major problem close to be solved is the concept of credal inference, i.e., the equivalent of statistical inference (in its Bayesian form) but within the TBM realm. The advantage will be that inference can be done with an a priori that really represents ignorance. Real life successful applications start to show up, essentially in the military domain, for object recognitions issues.

CONCLUSIONS

We have very shortly presented the TBM, a model for the representation of quantified beliefs based on belief functions. The whole theory has enormously increased since Shafer's seminal work. We only present very general ideas and provide pointers to the papers where the whole theory is presented. Full details must be found in the recent up to date literature.

REFERENCES

Ayoun, A. & Smets, P. (2001). Data association in multi-target detection using the transferable belief model. *International Journal of Intelligent Systems*, *16*, 1167-1182.

Ben Yaghlane, B., Smets, Ph., & Mellouli, K. (2001). Belief function independence: I. the marginal case. *International Journal of Approximate Reasoning*, *29*, 47-70.

Delmotte, F., & Smets, P., (2004). Target identification based on the transferable belief model interpretation of Dempster-Shafer model. *IEEE Trans. Syst., Man, Cybern. A., 34*, 457-471.

Denoeux, T., & Ben Yaghlane, A. (2002). Approximating the combination of belief functions using the fast mobius transform in a coarsened frame. *International Journal of Approximate Reasoning*, *31*, 77-101.

Denoeux, T., & Masson, M.-H. (2004). EVCLUS: Evidential clustering of proximity data. *IEEE Trans. SMC: B*, *34*, 95-109.

Elouedi, Z., Mellouli, K., & Smets, P. (2001). Belief decision trees: theoretical foundations. *International Journal of Approximate Reasoning*, *28*, 91-124.

Elouedi, Z., Mellouli, K., & Smets, P. (2004). Assessing sensor reliability for multisensor data fusion with the transferable belief model. *IEEE Trans. SMC B*, *34*, 782-787.

Haenni, R., & Lehmann, N. (2002). Resource-bounded and anytime approximation of belief function computations. *International Journal of Approximate Reasoning*, *32*, 103-154.

Lefevre, E., Colot, O., & Vannooremberghe, P. (2002). Belief functions combination and conflict management. *Information fusion*, *3*, 149-162

Milisavljevic, N., Bloch, I., & Acheroy, M. (2000). Modeling, combining and discounting mine detection sensors within Dempster- Shafer framework. In *Detection technologies for mines and minelike targets* (Vol. 4038, pp. 1461-1472). Orlando, USA: SPIE Press.

Shafer, G. (1976). *A mathematical theory of evidence*. Princeton, NJ: Princeton University Press.

Shenoy, P. P. (1997). Binary join trees for comput-

ing marginals in the Shenoy-Shafer architecture. *International Journal of Approximate Reasoning, 17*, 239-263.

Smets, P. (1998). The transferable belief model for quantified belief representation. In D. M. Gabbay & P. Smets (Eds.), *Handbook of defeasible reasoning and uncertainty management systems* (Vol. 1, pp. 267-301). The Netherlands: Kluwer, Doordrecht.

Smets, P. (2002a). Decision making in a context where uncertainty is represented by belief functions. In R. P. Srivastava, & T. J. Mock (Eds.), *Belief functions in business decisions* (pp. 17-61). Heidelberg, Germany: Physica-Verlag.

Smets, P. (2002b). The application of the matrix calculus to belief functions. *International Journal of Approximate Reasoning, 31*, 1-30.

Smets, P., & Kennes, R. (1994). The transferable belief model. *Artificial Intelligence, 66*, 191-234.

Smets, P., & Kruse, R. (1997). The transferable belief model for quantified belief representation. In A. Motro & P. Smets (Eds.), *Uncertainty in information systems: From needs to solutions* (pp. 343-368). Boston, MA: Kluwer.

KEY TERMS

Basic Belief Assignment: M(A) is the parts of belief that support that the actual world is in A without supporting any more specific subset of A.

Belief Function: Bel(A) is the total amount of belief that support that the actual world is in A without supporting its complement.

Conditioning: Revision process of a belief by a fact accepted as true.

Conjunctive Combination: The combination of the beliefs induced by several sources into an aggregated belief.

Open World Assumption: The fact that the frame of discernment might not be exhaustive.

Pignistic Probability Function: BetP is the probability function used for decision making.

Plausibility Function: Pl(A) is the total amount of belief that might support that the actual world is in A.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 1135-1139, copyright 2005 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

Tree and Graph Mining

Dimitrios Katsaros
Aristotle University, Greece

Yannis Manolopoulos
Aristotle Univeristy, Greece

INTRODUCTION

During the past decade, we have witnessed an explosive growth in our capabilities to both generate and collect data. Various data mining techniques have been proposed and widely employed to discover valid, novel and potentially useful patterns in these data. Data mining involves the discovery of patterns, associations, changes, anomalies, and statistically significant structures and events in huge collections of data.

One of the key success stories of data mining research and practice has been the development of efficient algorithms for discovering frequent itemsets – both sequential (Srikant & Agrawal, 1996) and non-sequential (Agrawal & Srikant, 1994). Generally speaking, these algorithms can extract co-occurrences of items (taking or not taking into account the ordering of items) in an efficient manner. Although the use of sets (or sequences) has effectively modeled many application domains, like market basket analysis, medical records, a lot of applications have emerged whose data models do not fit in the traditional concept of a set (or sequence), but require the deployment of richer abstractions, like graphs or trees. Such graphs or trees arise naturally in a number of different application domains including network intrusion, semantic Web, behavioral modeling, VLSI reverse engineering, link analysis and chemical compound classification.

Thus, the need to extract complex tree-like or graph-like patterns in massive data collections, for instance, in bioinformatics, semistructured or Web databases, became a necessity. The class of exploratory mining tasks, which deal with discovering patterns in massive databases representing complex interactions among entities, is called *Frequent Structure Mining* (FSM) (Zaki, 2002).

In this article we will highlight some strategic application domains where FSM can help provide significant results and subsequently we will survey the most important algorithms that have been proposed for mining graph-like and tree-like substructures in massive data collections.

BACKGROUND

As a motivating example for graph mining consider the problem of mining chemical compounds to discover recurrent (sub) structures. We can model this scenario using a graph for each compound. The vertices of the graphs correspond to different atoms and the graph edges correspond to bonds among the atoms. We can assign a label to each vertex, which corresponds to the atom involved (and maybe to its charge) and a label to each edge, which corresponds to the type of the bond (and maybe to information about the 3D orientation). Once these graphs have been generated, recurrent substructures become frequently occurring subgraphs. These graphs can be used in various tasks, for instance, in classifying chemical compounds (Deshpande, Kuramochi, & Karypis, 2003).

Another application domain where graph mining is of particular interest arises in the field of Web usage analysis (Nanopoulos, Katsaros, & Manolopoulos, 2003). Although various types of usage (traversal) patterns have been proposed to analyze the behavior of a user (Chen, Park, & Yu, 1998), they all have one very significant shortcoming; they are one-dimensional patterns and practically ignore the link structure of the site. In order to perform finer usage analysis, it is possible to look at the entire forward accesses of a user and to mine frequently accessed subgraphs of that site.

Looking for examples where tree mining has been successfully applied, we can find a wealth of them. A characteristic example is XML, which has been a very popular means for representing and storing information of various kinds, because of its modeling flexibility. Since tree-structured XML documents are the most

widely occurring in real applications, one would like to discover the commonly occurring subtrees that appear in the collections. This task could benefit applications, like database caching (Yang, Lee, & Hsu, 2003), storage in relational databases (Deutsch, Fernandez, & Suciu, 1999), building indexes and/or wrappers (Wang & Liu, 2000) and many more.

Tree patterns arise also in bioinformatics. For instance, researchers have collected large amounts of RNA structures, which can be effectively represented using a computer data structure called tree. In order to deduce some information about a newly sequenced RNA, they compare it with known RNA structures, looking for common topological patterns, which provide important insights to the function of the RNA (Shapiro & Zhang, 1990). Another application of tree mining in bioinformatics is found in the context of constructing phylogenetic trees (Shasha, Wang, & Zhang, 2004), where the task of phylogeny reconstruction algorithms it to use biological information about a set of e.g., taxa, in order to reconstruct an ancestral history linking together all the taxa in the set.

There are two distinct formulations for the problem of mining frequent graph (tree) substructures and are referred to as the *graph-transaction* (*tree-transaction*) setting and the *single-graph* (*single-tree*) setting. In the graph-transaction setting, the input to the pattern-mining algorithm is a set of relatively small graphs (called transactions), whereas in the single-graph setting the input data is a single large graph. The difference affects the way the frequency of the various patterns is determined. For the former, the frequency of a pattern is determined by the number of graph transactions that the pattern occurs in, irrespective of how many times a pattern occurs in a particular transaction, whereas in the latter, the frequency of a pattern is based on the number of its occurrences (i.e., embeddings) in the single graph. The algorithms developed for the graph-transaction setting can be modified to solve the single-graph setting, and vice-versa.

Depending also on the application domain, the considered graphs (trees) can be ordered or unordered, directed or undirected. No matter what these characteristics are, the (sub)graph mining problem can be defined as follows. (A similar definition can be given for the tree mining problem.) Given as input a database of graphs and a user-defined real number $0<\sigma\leq 1$, we need to find all frequent subgraphs, where the word "frequent" implies those subgraphs with frequency larger than or equal to the threshold σ. (In the following, equivalently to the term *frequency*, we use the term *support*.) We illustrate this problem for the case of graph-transaction, labeled, undirected graphs, with $\sigma=2/3$. The input and output of such an algorithm are given in Figure 1.

Although the very first attempts to deal with the problem of discovering substructure patterns from massive graph or tree data are dated back to the early 90's (Cook & Holder, 1994), only recently the field of mining for graph and tree patterns has flourished. A wealth of algorithms has been proposed, most of which are based on the original level-wise *Apriori* algorithm for mining frequent itemsets (Agrawal & Srikant, 1994). Next, we will survey the most important of them.

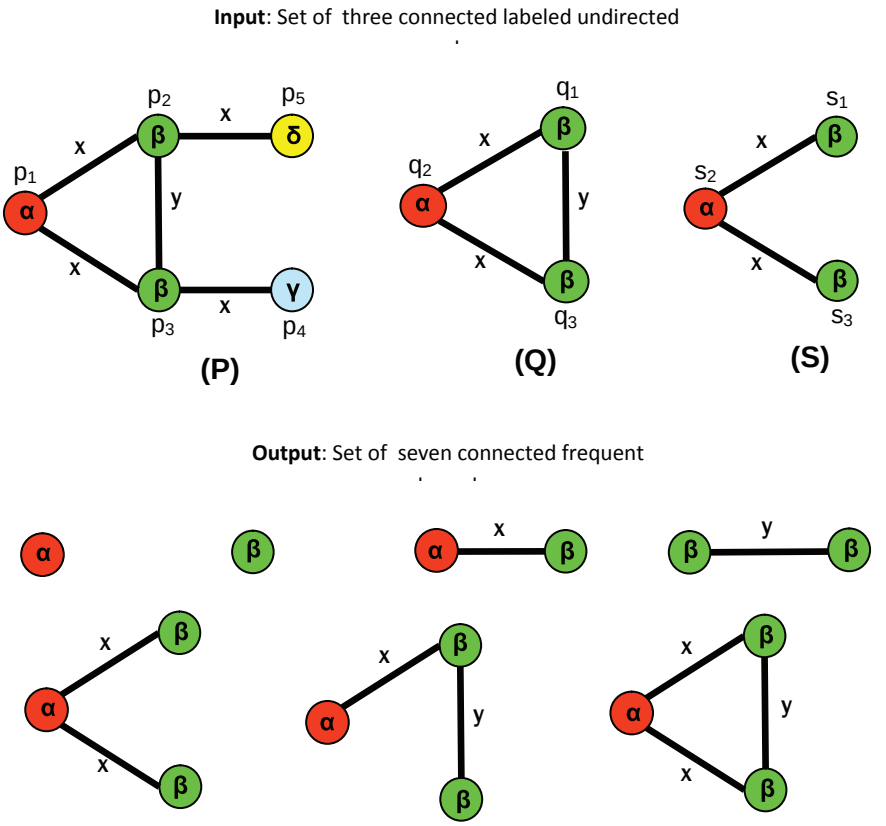
ALGORITHMS FOR GRAPH MINING

The graph is one of the most fundamental constructions studied in mathematics and thus, numerous classes of substructures are targeted by graph mining. These substructures include the *generic subgraph*, *induced subgraph*, *connected subgraph*, (ordered and unordered) *tree* and *path* (see Figure 2). We give the definitions of these substructures in the next paragraph and subsequently present the graph mining algorithms, able to discover all frequent substructures of any kind mentioned earlier.

Following mathematical terminology, a graph is represented as G(V, E, f), where V is a set of vertices, E is a set of edges connecting pairs of vertices and f is a function $f:E\rightarrow VxV$. For instance, in Figure 2 we see that $f(e_1)=(v_1,v_2)$. We say that *GS*(Vs,Es,f) is a *generic subgraph* of G, if $Vs\subset V$, $Es\subset E$ and $v_i,v_j\in Vs$ for all edges $f(e_k)=(v_i,v_j)\in Es$. An *induced subgraph ISG*(Vs,Es,f) of G has a subset of vertices of G and the same edges between pairs of vertices as in G, in other words, $Vs\subset V$, $Es\subset E$ and $\forall v_i,v_j\in Vs$, $e_k=(v_i,v_j)\in Es\Leftrightarrow f(e_k)=(v_i,v_j)\in E$. We say that *CSG*(Vs,Es,f) is a connected subgraph of G, if $Vs\subset V$, $Es\subset E$ and all vertices in Vs are reachable through some edges in Es. An acyclic subgraph of G is called a *tree T*. Finally, a tree of G which does not include any braches is a *path P* in G.

The first algorithm for mining all frequent subgraph patterns is *AGM* (Inocuchi, Washio, & Motoda, 2000, 2003). AGM can mine various types of patterns, namely generic subgraphs, induced subgraphs, connected subgraphs, ordered and unordered trees and subpaths.

Figure 1. Mining frequent labeled undirected graphs.



AGM is based on a labeling of the graph's vertices and edges, a "canonical labeling" (Washio & Motoda, 2003) for the adjacency matrix of the graph, which allows for the unambiguous representation of the graph. The basic principle of AGM is similar to that of Apriori for basket analysis. Starting from frequent graphs, where each graph is a single vertex, the frequent graphs having larger sizes are searched in bottom up manner by generating candidates having an extra vertex.

A similar canonical labeling and the Apriori-style is also followed by *FSG* (Kuramochi & Karypis, 2004b), by its variations like *gFSG* (2002), *HSIGRAM* and *VSIGRAM* (2004), and by the algorithm proposed in (Vanetik, Gudes, & Shimony, 2002).

Finally, based on the same principles as above, (Huan, Wang, & Prins, 2003) proposed an interesting labeling for graphs based on the adjacency matrix representation, which can drastically reduce the computation complexity of the candidate generation phase.

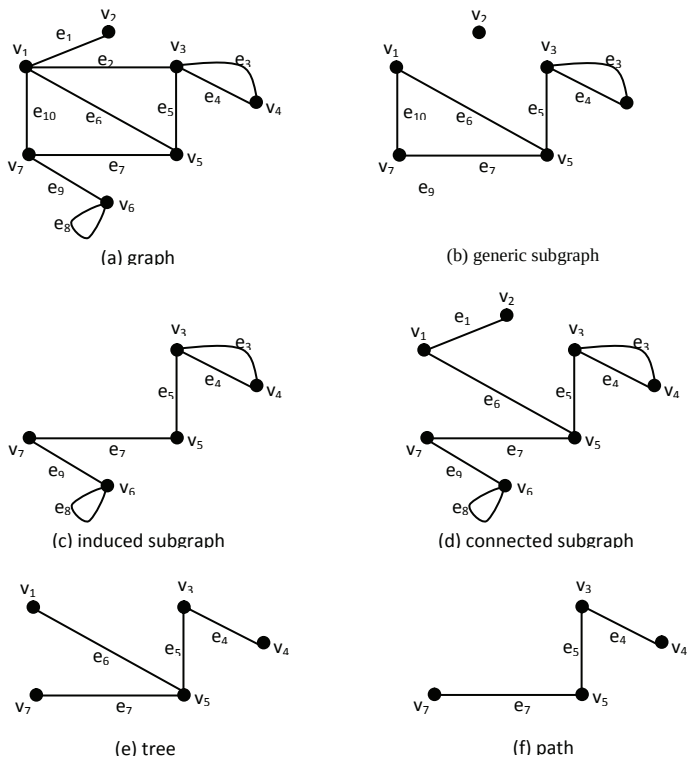
Recently, a depth-first-searching (DFS) "canonical labeling" was introduced by *gSpan* (Yan & Han, 2002) and its variation for mining closed frequent graph patterns, the *CloseGraph* (2003). The main difference of their coding from other approaches is that they use a tree representation of each graph instead of the adjacency matrix to define the code of the graph.

ALGORITHMS FOR TREE MINING

There has been very little work in mining all frequent subtrees; only a handful of algorithms can be found in the literature. Although graph mining algorithms can be applied to this problem as well, they are likely to be too general, since they do not exploit the existence of the root and the lack of cycles. Thus, they are most probably very inefficient.

The first effort towards tree mining was conducted by Ke Wang (Cong, Yi, Liu, & Wang, 2002; Wang & Liu, 2000). Their algorithm is an application of the original level-wise Apriori to the problem of mining frequently occurring collections of paths, where none of them is a prefix of the other and thus they

Figure 2. Characteristic graph substructures.



correspond to trees. Later, (Katsaros, Nanopoulos, & Manolopoulos, 2004) developed a numbering scheme for labeled, ordered trees in order to speed-up the execution of Wang's algorithm, and (Katsaros, 2003) proposed the *DeltaSSD* algorithm to study the efficient maintenance of the discovered tree substructures under database updates.

Departing from the Apriori paradigm, (Zaki, 2002) proposed the *TREEMINER* and (Abe, Kawasoe, Asai, Arimura, & Arikawa, 2002; Asai et al., 2002; Asai, Arimura, Uno, & Nakamo, 2003) proposed the *FREQT* algorithms. These tree mining algorithms are very similar; they are both based on an efficient enumeration technique that allows for the incremental construction of the set of frequent tree patterns and their occurrences level by level. TREEMINER performs a depth-first search for frequent subtrees and uses a vertical representation for the trees in the database for fast support counting. FREQT uses the notion of rightmost expansion to generate candidate trees by attaching nodes only to the rightmost branch of a frequent subtree. Similar, in spirit are the algorithms proposed by (Chi, Yang, & Muntz, 2003, 2004). Finally, Xiao, Yao, Li, and Dunham (2003) proposed a method for discovering only the maximal frequent subtrees using a highly condensed data structure for the representation and finding of the maximal frequent trees of the database.

The tree mining problem is also related to the *tree isomorphism* and *tree pattern matching* problems. Though, the fundamental difference between the tree mining and the other two research problems is the fact that the tree mining algorithms focus on discovering *all* frequent tree substructures, and not only deciding whether or not a particular instance of a tree is "contained" in another larger tree.

FUTURE TRENDS

The field of tree and graph mining is still in its infancy. To date research and development have focused on the discovery of very simple structural patterns. Although, it is difficult to foresee the future directions in the field, because they depend on the application requirements, we believe that the discovery of richer patterns requires new algorithms. For instance, algorithms are needed

for the mining of weighted trees or graphs. Moreover, the embedding of the tree and graph mining algorithms into clustering algorithms will be a very active area for both research and practice.

CONCLUSION

During the past decade, we have witnessed the emergence of the data mining field as a novel research area, investigating interesting problems and developing real-life applications. Initially, the targeted data formats were limited to relational tables, comprised by unordered collections of rows, where each row of the table is treated as a set. Due to the rapid proliferation of many applications from biology, chemistry, bioinformatics and communication networking, whose data model can only be described by a graph or a tree, the research field of *graph* and *tree mining* started to emerge. The field provides many attractive topics for both theoretical and engineering achievements and it is expected to be one of the key fields in data mining research for the years ahead.

REFERENCES

Abe, K., Kawasoe, S., Asai, T., Arimura, H., & Arikawa, S. (2002). Optimized substructure discovery for semi-structured data. *Proceedings of the European Conference on Principles of Data Mining and Knowledge Discovery (PKDD)*, Lecture Notes in Artificial Intelligence (LNAI), vol. 2431, 1-14.

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules in large databases. *Proceedings of the International Conference on Very Large Data Bases (VLDB)* (pp. 487-499).

Asai, T., Abe, K., Kawasoe, S., Arimura, H., Sakamoto, H., & Arikawa, S. (2002). Efficient substructure discovery for large semi-structured data. *Proceedings of the SIAM Conference on Data Mining (SDM)* (pp. 158-174).

Asai, T., Arimura, H., Uno, T., & Nakano, S. (2003). Discovering frequent substructures in large unordered trees. *Proceedings of the Conference on Discovery Sciences (DS)*, Lecture Notes in Artificial Intelligence (LNAI), vol. 2843, 47-61.

Chen, M.S., Park, J.S., & Yu, P.S. (1998). Efficient data mining for path traversal patterns. *IEEE Transactions on Knowledge and Data Engineering, 10*(2), 209-221.

Chi, Y., Yang, Y., & Muntz, R.R. (2003). Indexing and mining free trees. *Proceedings of the IEEE International Conference on Data Mining (ICDM)* (pp. 509-512).

Chi, Y., Yang, Y., & Muntz, R.R. (2004). HybridTreeMiner: An efficient algorithm for mining frequent rooted trees and free trees using canonical forms. *Proceedings of the IEEE Conference on Scientific and Statistical Data Base Management (SSDBM)* (pp. 11-20).

Cong, G., Yi, L., Liu, B., & Wang, K. (2002). Discovering frequent substructures from hierarchical semi-structured data. *Proceedings of the SIAM Conference on Data Mining (SDM)* (pp. 175-192).

Cook, D.J., & Holder, L.B. (1994). Substructure discovery using minimum description length and background knowledge. *Journal of Artificial Intelligence Research, 1*, 231-255.

Deshpande, M., Kuramochi, M., & Karypis, G. (2003). Frequent sub-structure-based approaches for classifying chemical compounds. *Proceedings of the IEEE International Conference on Data Mining (ICDM)* (pp. 35-42).

Deutsch, A., Fernandez, M.F., & Suciu, D. (1999). Storing semistructured data with STORED. *Proceedings of the ACM International Conference on Management of Data (SIGMOD)* (pp. 431-442).

Huan, J., Wang, W., & Prins, J. (2003). Efficient mining of frequent subgraphs in the presence of isomorphism. *Proceedings of the IEEE International Conference on Data Mining (ICDM)* (pp. 549-552).

Inokuchi, A., Washio, T., & Motoda, H. (2000). An apriori-based algorithm for mining frequent substructures from graph data. *Proceedings of the European Conference on Principles of Data Mining and Knowledge Discovery (PKDD)*, Lecture Notes in Artificial Intelligence (LNAI), vol. 1910, 13-23.

Inokuchi, A., Washio, T., & Motoda, H. (2003). Complete mining of frequent patterns from graphs: Mining graph data. *Machine Learning, 50*(3), 321-354.

Katsaros, D. (2003). Efficiently maintaining structural associations of semistructured data. *Lecture Notes on Computer Science (LNCS)*, vol. 2563,118-132.

Katsaros, D., Nanopoulos, A., & Manolopoulos, Y. (2005). Fast mining of frequent tree structures by hashing and indexing. *Information & Software Technology, 47*(2), 129-140.

Kuramochi, M., & Karypis G. (2002). Discovering Frequent Geometric Subgraphs, *Proceedings of the IEEE International Conference on Data Mining (ICDM)* (pp. 258-265).

Kuramochi, M., & Karypis G. (2004). Finding frequent patterns in a large sparse graph. *Proceedings of the SIAM Conference on Data Mining (SDM).*

Kuramochi, M., & Karypis G. (2004b). An efficient algorithm for discovering frequent subgraphs. *IEEE Transactions on Knowledge and Data Engineering, 16*(9), 1038-1051.

Nanopoulos, A., Katsaros, D., & Manolopoulos, Y. (2003). A data mining algorithm for generalized web prefetching. *IEEE Transactions on Knowledge and Data Engineering, 15*(5), 1155-1169.

Shapiro, B., & Zhang, K. (1990). Comparing multiple RNA secondary structures using tree comparisons. *Computer Applications in Biosciences, 6*(4), 309-318.

Shasha, D., Wang, J.T.L., & Zhang, S. (2004). Unordered tree mining with applications to phylogeny. *Proceedings of the IEEE International Conference on Data Engineering (ICDE)* (pp. 708-719).

Srikant, R., & Agrawal, R. (1996). Mining sequential patterns: Generalizations and performance improvements. *Proceedings of the International Conference on Extending Database Technology (EDBT'96)* (pp. 3-17).

Vanetik, N., Gudes, E., & Shimony, S.E. (2002). Computing frequent graph patterns from semistructured data. *Proceedings of the IEEE International Conference on Data Mining (ICDM)* (pp. 458-465).

Wang, K., & Liu, H. (2000). Discovering structural association of semistructured data. *IEEE Transactions on Knowledge and Data Engineering, 12*(3), 353-371.

Washio, T. & Motoda, H. (2003). State of the art of graph-based Data Mining. *ACM SIGKDD Explorations, 5*(1), 59-68.

Xiao, Y., Yao, J.-F., Li, Z., & Dunham, M.H. (2003). Efficient data mining for maximal frequent subtrees. *Proceedings of the IEEE International Conference on Data Mining (ICDM)* (pp. 379-386).

Yan, X., & Han, J. (2002). gSpan: Graph-based substructure pattern mining. *Proceedings of the IEEE International Conference on Data Mining (ICDM)* (pp. 721-724).

Yan, X., & Han, J. (2003). CloseGraph: Mining closed frequent graph patterns. *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)* (pp. 286-295).

Yang, L.H., Lee, M.L., & Hsu, W. (2003). Efficient mining of XML query patterns for caching. *Proceedings of the International Conference on Very Large Data Bases (VLDB)* (pp. 69-80).

Zaki, M. (2002). Efficiently mining frequent trees in a forest. *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)* (pp. 71-80).

KEY TERMS

Closed Frequent Graph: A frequent graph pattern G is *closed* if there exists no proper super-pattern of G with the same support in the dataset.

Correlation: describes the strength or degree of linear relationship. That is, correlation lets us specify to what extent the two variables behave alike or vary together. Correlation analysis is used to assess the simultaneous variability of a collection of variables. For instance, suppose one wants to study the simultaneous changes with age of height and weight for a population. Correlation analysis describes the how the change in height can influence the change in weight.

Embedded Subtree: Let T(N,B) be a tree, where N represents the set of its nodes and B the set of its edges. We say that a tree S(Ns,Bs) is an *embedded subtree* of T provided that: i) $Ns \subseteq N$, ii) $b=(n_x,n_y) \in Bs$ if and only if n_x is an ancestor of n_y in T. In other words, we require that a branch appear in S if and only if the two vertices are on the same path from the root to a leaf in T.

Exploratory Data Analysis (EDA): comprises a set of techniques used to identify systematic relations between variables when there are no (or not complete) a priori expectations as to the nature of those relations.

In a typical exploratory data analysis process, many variables are taken into account and compared, using a variety of techniques in the search for systematic patterns.

Free Tree: Let G be a connected acyclic labeled graph. If we label the leaves (because of its acyclicity, each connected acyclic labeled graph has at least one node which is connected to the rest of the graph by only one edge, that is, a leaf) with zero and the other nodes recursively with the minimal label of its neighbors plus one, then we get an unordered, unrooted tree-like structure, a so-called *free tree*. It is a well-known fact that every free tree has at most two nodes, which minimize the maximal distance to all other nodes in the tree, the so-called *centers*.

Induced Subtree: Let T(N,B) be a tree, where N represents the set of its nodes and B the set of its edges. We say that a tree S(Ns,Bs) is an *induced subtree* of T provided that: i) $Ns \subseteq N$, ii) $b=(n_x,n_y) \subseteq Bs$ if and only if n_x is a parent of n_y in T. Thus, induced subtrees are a specialization of embedded subtrees.

Linear Regression: is used to make predictions about a single value. Simple linear regression involves discovering the equation for a line that most nearly fits the given data. That linear equation is then used to predict values for the data. For instance, if a cost modeler wants to know the prospective cost for a new contract based on the data collected from previous contracts, then s/he may apply linear regression.

Uncertainty Operators in a Many-Valued Logic

Herman Akdag
University Paris6, France

Isis Truck
University Paris8, France

INTRODUCTION

This article investigates different tools for knowledge representation and modelling in decision making problems. In this variety of AI systems the experts' knowledge is often heterogeneous, that is, expressed in many forms: numerical, interval-valued, symbolic, linguistic, etc. Linguistic concepts (adverbs, sentences, sets of words…) are sometimes more efficient in many expertise domains rather than precise, interval-valued or fuzzy numbers. In these cases, the nature of the information is qualitative and the use of such concepts is appropriate and usual. Indeed, in the case of fuzzy logic for example, data are represented through fuzzy functions that allow an infinite number of truth values between 0 and 1. Instead, it can be more appropriate to use a finite number of qualitative symbols because, among other reasons, any arbitrary fuzzification becomes useless; because an approximation will be needed at the end anyway; etc. A deep study has been recently carried out about this subject in (Gottwald, 2007).

In this article we propose a survey of different tools manipulating these symbols as well as human reasoning handles with natural linguistic statements. In order to imitate or automatize expert reasoning, it is necessary to study the representation and handling of discrete and linguistic data (Truck & Akdag, 2005; Truck & Akdag, 2006). One representation is the *many-valued logic* framework in which this article is situated.

The many-valued logic, which is a generalization of classical boolean logic, introduces truth degrees which are intermediate between *true* and *false* and enables the partial truth notion representation. There are several many-valued logic systems (Lukasiewicz's, Gödel's, etc.) comprising finite-valued or infinite-valued sets of truth degrees. The system addressed in this article is specified by the use of $\mathcal{L}_M = \{\tau_0,...,\tau_i,...,\tau_{M-1}\}$ [1] a totally ordered finite set[2] of truth-degrees ($\tau_i \leq \tau_j \Leftrightarrow i \leq j$) between τ_0 (false) and τ_{M-1} (true), given the operators $\vee$ (max), $\wedge$ (min) and $\neg$ (negation or symbolic complementation, with $\neg\tau_j = \tau_{M-j-1}$) and the following Lukasiewicz implication $\rightarrow_L$: $\tau_i \rightarrow_L \tau_j = \min(\tau_{M-1}, \tau_{M-1-(i-j)})$

These degrees can be seen as membership degrees: x partially belongs to a multiset[3] A with a degree τ_i if and only if $x \in_{\tau_i} A$. The many-valued logic presented here deals with linguistic statements of the following form: x is v_α A where x is a variable, v_α a scalar adverb (such as "very", "more or less", etc.) and A a gradable linguistic predicate (such as "tall", "hot", "young"...). The predicate A is satisfiable to a certain degree expressed through the scalar adverb v_α. The following interpretation has been proposed (Akdag, De Glas & Pacholczyk, 1992):

x is v_α $A \Leftrightarrow$ "x is A" is τ_α – true

Qualitative degrees constitute a good way to represent uncertain and not quantified knowledge, indeed they can be associated with Zadeh's linguistic variables (Zadeh, 2004) that model approximate reasoning well. Using this framework, several qualitative approaches for uncertainty representation have been presented in the literature. For example, in (Darwiche & Ginsberg, 1992; Seridi & Akdag, 2001) the researchers want to find a model which simulates cognitive activities, such as the management of uncertain statements of natural language that are defined in a finite totally ordered set of symbolic values. The approach consists in representing and exploiting the uncertainty by qualitative degrees, as probabilities do with numerical values. In order to manipulate these symbolic values, four elementary operators are outlined: multiplication, addition, subtraction and division (Seridi & Akdag, 2001). Then two other kinds of operators are given: modification tools based on scales and symbolic aggregators.

BACKGROUND

Extending the work of Akdag, De Glas & Pacholczyk on the representation of uncertainty via the many-valued logic (Akdag, De Glas & Pacholczyk, 1992), several studies have been led where an axiomatic system for symbolic probability theory has been proposed (Seridi & Akdag, 2001; Khayata, Pacholczyk & Garcia, 2002).

The qualitative uncertainty theory we present here takes place between the classical probability theory and possibility theory. The idea is to translate the four basic operations respecting required properties with well-chosen formulas. The qualitative and the numerical models are linked together using the four symbolic operators that compute the qualitative operations.

These works related to the qualitative uncertainty theory have in common the following points:

- They are based on the association "probability degrees / logic": In their work, laws of probabilities are obtained thanks to logical operators.
- They have developed an axiomatic theory, which allows obtaining results either from axioms, or from theorems.
- In addition, they permit the use of uncertainty (and imprecision) expressed in the qualitative form.

The considered qualitative degrees of uncertainty belong to the graduated scale $\mathcal{L}_M$. The first step is to introduce a total order in the scale of degrees. Then in order to be able to translate in symbolic the different axioms and theorems of the classical probability theory, three elementary operators must be defined as in (Darwiche & Ginsberg, 1992). Indeed a symbolic addition (or a symbolic t-conorm, to generalize the addition), a symbolic multiplication (or a symbolic t-norm in order to be able to translate the disconditioning and the independence) and a symbolic division (to translate the conditioning) must be provided.

Moreover, another constraint is introduced: "*it is necessary that if C is the result of the division of A by B then B multiplied by C gives A*" (relationship between the qualitative multiplication and the qualitative division). This intuitive constraint has been proposed for the first time in (Seridi & Akdag, 2001). Another originality lies in the definition of a qualitative subtraction instead of the use (sometimes artificially) of the complementation operator. Thus, a symbolic difference and a symbolic distance are defined to translate both the subtraction and the absolute value of the subtraction.

Formulas for Uncertainty Qualitative Theory

A qualitative multiplication of two degrees τ_α and τ_β is defined by the function MUL from $\mathcal{L}_M \times \mathcal{L}_M$ to $\mathcal{L}_M$ that verifies the properties of a t-norm to which are added the absorbent element τ_0 and the complementarity property: $\text{MUL}(\tau_\alpha,\neg\tau_\alpha) = \tau_0$.

Similarly, a qualitative addition of two degrees τ_α and τ_β is a function ADD from $\mathcal{L}_M \times \mathcal{L}_M$ to $\mathcal{L}_M$ that verifies the properties of a t-conorm to which are added the absorbent element τ_{M-1} and the complementarity property: $\text{ADD}(\tau_\alpha,\neg\tau_\alpha) = \tau_{M-1}$.

The qualitative subtraction of two degrees τ_α and τ_β such that $\tau_\beta \le \tau_\alpha$ is defined by the function SOUS from $\mathcal{L}_M \times \mathcal{L}_M$ to $\mathcal{L}_M$ that verifies the following properties: increasing relatively to the first argument; decreasing relatively to the second argument; SOUS has a neutral element τ_0 and the subtraction of two identical degrees gives the neutral element.

SOUS allows us to define an important axiom linking probability of the union to the probability of the intersection (see U6 below). SOUS corresponds in fact to the bounded difference of Zadeh, defined in the fuzzy logic framework.

The qualitative division of two degrees τ_α and τ_β and such that $\tau_\alpha \le \tau_\beta$ with $\tau_\beta \neq \tau_0$ is defined by the function DIV from $\mathcal{L}_M \times \mathcal{L}_M$ to $\mathcal{L}_M$ that verifies the following properties: increasing relatively to the first argument; decreasing relatively to the second argument; DIV has an absorbent element τ_0 and a neutral element τ_{M-1} and the division of two identical degrees, except for the absorbent element, gives the neutral element (boundary conditions).

Our choices for the four operators are the following:

1. $\text{MUL}_{\mathcal{L}}(\tau_\alpha,\tau_\beta) = \neg(\tau_\alpha \rightarrow_L \neg\tau_\beta) = \tau_\gamma$ therefore $\gamma = \max(\alpha+\beta-(M-1),0)$

2. $\text{ADD}_{\mathcal{L}}(\tau_\alpha,\tau_\beta) = (\neg\tau_\alpha \rightarrow_L \tau_\beta) = \tau_\delta$ therefore $\delta = \min(\alpha+\beta,M-1)$

3. $\text{SOUS}_{\mathcal{L}}(\tau_\alpha,\tau_\beta) = \neg(\tau_\alpha \rightarrow_L \tau_\beta) = \tau_S$ therefore $S = \max(\alpha-\beta,0)$

The 4th Operator

$$\begin{cases} \text{DIV}_{\mathcal{L}}(\tau_0,\tau_\beta) = \text{MUL}_{\mathcal{L}}(\tau_0,\tau_\beta) \text{ if } \tau_\beta \neq \tau_0 \\ \text{DIV}_{\mathcal{L}}(\tau_\alpha,\tau_\beta) = \text{ADD}_{\mathcal{L}}(\tau_\alpha,\neg\tau_\beta) \text{ if } \tau_\alpha \neq \tau_0 \text{ and } \tau_\beta \neq \tau_0 \text{ and } \tau_\alpha \leq \tau_\beta \\ \text{otherwise } \text{DIV}_{\mathcal{L}}(\tau_\alpha,\tau_\beta) \text{ is undefined} \end{cases}$$

We shall also define a fifth operator corresponding to the distance and specified according to SOUS:

$$\text{DIST}_{\mathcal{L}}(\tau_\alpha,\tau_\beta) = \begin{cases} \text{SOUS}_{\mathcal{L}}(\tau_\alpha,\tau_\beta) & \text{if } \tau_\alpha \geq \tau_\beta \\ \text{SOUS}_{\mathcal{L}}(\tau_\beta,\tau_\alpha) & \text{otherwise} \end{cases}$$

Axiomatic System for Qualitative Uncertainty Theory

Inside this qualitative uncertainty concept, a new axiomatic system has been given in (Seridi & Akdag, 2001) by using the predicate ***Inc*** of the many-valued logic. Thus, the proposition "*A* is τ_α–certain" is translated into $Inc(A) = \tau_\alpha$. The first axioms[4] are those proposed in (Akdag, De Glas & Pacholczyk, 1992):

U1. If $\models_{\tau_{M-1}} A$ then $Inc(A) = \tau_{M-1}$
i.e. if *A* is a tautology then *A* is certain
U2. If $\models_{\tau_0} A$ then $Inc(A) = \tau_0$
i.e. if *A* is false then *A* is impossible
U3. If $\models_{\tau_\alpha} A$ then $Inc(\neg A) = \neg\tau_\alpha$
i.e. if *A* is τ_α–certain then $\neg A$ is $\neg\tau_\alpha$–certain
U4. If $A \equiv B$ then $Inc(A) = Inc(B)$
i.e. if *A* is equivalent to *B* then $Inc(A) = Inc(B)$

This permits to have trivial theorems:

T1. If $Inc(A) = \tau_{M-1}$ then $Inc(\neg A) = \tau_0$
T2. If $Inc(A) = \tau_0$ then $Inc(\neg A) = \tau_{M-1}$

Definition. Let *A* and *B* be two quasi-independent events. If $Inc(A) = \tau_\alpha$ and $Inc(B) = \tau_\beta$ then $Inc(A \cap B) = \tau_\gamma$ where $\tau_\gamma = \text{MUL}_{\mathcal{L}}(\tau_\alpha,\tau_\beta)$

We obtain the following axioms:

U5. If $Inc(A) = \tau_\alpha$ and $Inc(B) = \tau_\beta$ and $Inc(A \cap B) = \tau_0$
then $Inc(A \cup B) = \text{ADD}_{\mathcal{L}}(\tau_\alpha,\tau_\beta)$
U6. If $Inc(A) = \tau_\alpha$ and $Inc(B) = \tau_\beta$ and $B \subset A$
then $Inc(A \cap \neg B) = \text{SOUS}_{\mathcal{L}}(\tau_\alpha,\tau_\beta)$
U7. If $Inc(A) = \tau_\alpha$ and $Inc(B) = \tau_\beta$ and $Inc(A \cap B) = \tau_\gamma$
then $Inc(A \cup B) = \text{ADD}_{\mathcal{L}}(\text{SOUS}_{\mathcal{L}}(\tau_\alpha,\tau_\gamma),\tau_\beta)$
U8. If $Inc(A \cap B) = \tau_\gamma$ and $Inc(A) = \tau_\alpha$
then $Inc(B/A) = \text{DIV}_{\mathcal{L}}(\tau_\gamma,\tau_\alpha)$

A particularity of axiom (U7) is that it is necessary to perform first the subtraction then the addition. The goal of this axiomatic system is to show that the chosen symbolic operators permit to demonstrate properties that can be compared to the classical results of probability theory (Seridi & Akdag, 2001). The approach proposed for the representation and management of qualitative uncertainty is in agreement with the intuition, especially through the four symbolic arithmetic operations. Moreover, with its axiomatic of symbolic probabilities, this model constitutes an interesting alternative to the classical probability theory. Advanced theorems of this axiomatic (Truck & Akdag, 2006) permit to manipulate Bayesian networks and to manage uncertainties in Knowledge-Based Systems in case of lack of reliability quantitative estimation. In other terms, they allow us to reason under uncertainty or imprecision. An interesting application of these operators has been proposed in (Revault d'Allonnes, Akdag & Poirel, 2007) where the authors use them to qualify and evaluate an information's certainty based on a confirmation criterion and weighted by its source's credibility.

MAIN FOCUS

In many practical cases, a fixed number of degrees is used (e.g. the set of seven degrees, $\mathcal{L}_7$). Thus, in a reasoning process, some results of a cognitive operator

will be expressible and some others won't, especially when the result is between two consecutive degrees. In this last case, an approximation is needed. Indeed, one cannot express a piece of knowledge with more (or less) precision than the granularity of the range, at this stage. This point is treated in (Truck & Akdag, 2006) where we propose tools to handle this problem.

The issue of data modification has been addressed by several authors. See for example (Bouchon-Meunier & Marsala, 2001; López de Mántaras & Arcos, 2002; Truck, Borgi & Akdag, 2002; Zadeh, 2004; etc.). Zadeh has introduced the notion of linguistic hedges and defined reinforcing and weakening fuzzy modifiers. This notion of modifiers is useful in many fields where the perception plays an important role. For example, in the context of music, López de Mántaras & Arcos have presented a system able to generate expressive music by means of fuzzy modification of parameters (e.g. vibrato, dynamics, rubato...). In the context of color, Aït Younes et al. use fuzzy subsets and modifiers to obtain certain colors in queries for image retrieval (Aït Younes, Truck & Akdag, 2007).

Towards Scales

In another context, Herrera et al. have proposed a pair to represent the modification between two fuzzy subsets (Herrera, Martínez & Sánchez, 2005). This pair composed of a *linguistic term* and a *translation* depends on a certain *hierarchy* or precision level inside a scale of non-uniformly distributed symbols. Several hierarchies are used to express the distribution of symbols in a fuzzy way. In the many-valued logic context, another approach of data modification using uniformly distributed scales has been proposed (Akdag, Truck, Mellouli & Borgi, 2001). These scales take place in the measure theory. Usually the measure scales are divided into four classes: nominal, ordinal, interval and ratio scales (Grabisch, 2001). A nominal scale does not express any value or relationship between variables (except equality). An ordinal scale is a scale on which data is shown simply in order of magnitude. These scales permit the measurement of degrees of difference, but not the specific amount of difference. An interval scale is a scale according to which the differences between values can be quantified in absolute but not relative terms and for which any zero is merely arbitrary. A ratio scale permits the comparison of differences of values. It has a fixed zero value. In (Akdag, Truck, Borgi & Mellouli, 2001) we have proposed *qualitative modifiers* that permit the refinement of a symbolic variable. These modifiers act on degrees as well as on scales themselves and they permit to modify data at will, i.e. to reach the needed level of precision. The scales we use are interval scales since there is no fixed zero value and differences of values cannot always be compared.

The Generalized Symbolic Modifiers

Going from one degree to another implies a change of both the degree and the scale. In particular, it implies a dilation and/or erosion of scales. Thus this permits to obtain modifiers that are functions allowing us to go from a degree to another. These *linguistic symbolic modifiers* have been formalized in (Akdag, Truck, Borgi & Mellouli, 2001): the *generalized symbolic modifiers* (GSMs). Information is converted into sets of ordered degrees attached to variables. The GSMs deal with linguistic expressions such as "much more than"... Three families of GSMs have been proposed: weakening, reinforcing and centring ones. They are characterized through formal definitions and entail an order relation. Thus, they can be presented in a lattice. Moreover, in order to express any kind of modification, it is possible to combine GSMs: mathematical compositions of some specific families of GSMs with others are proposed in (Truck & Akdag, 2005).

In the aforementioned studies, the symbolic degrees all have the same importance. But in many cases, such as opinion polls, weights associated to degrees are usually not the same. Weights may express the confidence or significance level of a judgment. The challenge is to give the most representative judgment.

The Symbolic Weighted Median

In (Truck, Akdag & Borgi, 2003) we have proposed a tool to answer this question: the *symbolic weighted median* (SWM) that gives the emergent element of a set of ordered weighted degrees. The introduced operator is constructed using the generalized symbolic modifiers. Indeed, the SWM gives as a representative element, an element from the initial set of weighted degrees, more or less *modified*. This means that GSMs and SWM are closely related. The SWM satisfies the properties of identity, monotonicity, symmetry, idempotence, compensation, boundary conditions, continuity and counterbalancement.

As mentioned above, the compositions of GSMs permit to express linguistic assertions composed of more than one adverb. Moreover, a certain composition of GSMs can express the result of an aggregation. This explains why computation of the SWM is expressed in terms of the GSMs. When computing the SWM with the GSMs, an association is implicitly done between modifiers and aggregation. Actually this kind of aggregation can be seen as finding a *modification* of an element from the initial set.

The Symbolic Average

Many other kinds of combination operators exist: for example, arithmetic mean, geometric mean, standard deviation... so, according to the same model than above we consider a symbolic average aggregator based on the classical average: the Symbolic Average (SA). In this article, we only focus on a binary SA (BSA) that merges two truth degrees.

According to Dubois & Prade, an averaging operator (or a mean) is a function $\mathcal{A} : [0,1] \times [0,1] \rightarrow [0,1]$ that satisfies (Dubois & Prade, 1985):

i. $x \wedge y \leq \mathcal{A}(x,y) \leq x \vee y$
ii. $\mathcal{A}(x,y) = \mathcal{A}(y,x)$ (commutativity)
iii. $\mathcal{A}$ is increasing and continuous
iv. $\mathcal{A}(x,x) = x$ (idempotency)

Thus in our context the BSA is a function AVE from $\mathcal{L}_M \times \mathcal{L}_M$ to $\mathcal{L}_M$ that verifies the properties above and also the following:

$$\text{DIST}(\text{SOUS}(\tau_i,\text{AVE}(\tau_i,\tau_j)),\text{SOUS}(\text{AVE}(\tau_i,\tau_j)),\tau_j) \leq \tau_1$$

Let $\text{AVE}(\tau_i,\tau_j) = \tau_\mu$. We propose two BSAs ($\text{AVE}_{\mathcal{L}}^{\text{low}}$ and $\text{AVE}_{\mathcal{L}}^{\text{up}}$):

- ***Lower BSA (denoted* $\mathbf{AVE}_{\mathcal{L}}^{\mathbf{low}}$*).***

$\text{SOUS}_{\mathcal{L}}(\tau_i,\text{AVE}_{\mathcal{L}}(\tau_i,\tau_j)) = \text{SOUS}_{\mathcal{L}}(\text{AVE}_{\mathcal{L}}(\tau_i,\tau_j),\tau_j)$ or $\text{SOUS}_{\mathcal{L}}(\tau_i,\text{AVE}_{\mathcal{L}}^{\text{low}}(\tau_i,\tau_j)) = \text{SOUS}_{\mathcal{L}}(\text{AVE}_{\mathcal{L}}^{\text{up}}(\tau_i,\tau_j),\tau_j)$.
Thus $\mu = \lfloor (i+j-1)/2 \rfloor$

- ***Upper BSA (denoted* $\mathbf{AVE}_{\mathcal{L}}^{\mathbf{up}}$*).***

$\text{SOUS}_{\mathcal{L}}(\tau_i,\text{AVE}_{\mathcal{L}}(\tau_i,\tau_j)) = \text{SOUS}_{\mathcal{L}}(\text{AVE}_{\mathcal{L}}(\tau_i,\tau_j),\tau_j)$ or $\text{SOUS}_{\mathcal{L}}(\tau_i,\text{AVE}_{\mathcal{L}}^{\text{up}}(\tau_i,\tau_j)) = \text{SOUS}_{\mathcal{L}}(\text{AVE}_{\mathcal{L}}^{\text{low}}(\tau_i,\tau_j)),\tau_j)$.
Thus $\mu = \lfloor (i+j+1)/2 \rfloor$

We notice that the BSA is equivalent to the SWM when the two truth degrees' weights are equal and when an approximation is *not* needed. It should be advisable that the SA (generalization of the BSA) when computed with more than two degrees, give results different from those of the SWM.

FUTURE TRENDS

Inside this many-valued logic, it would be interesting to provide a complete panel of qualitative operators in order to be able to propose several tools for reasoning under imprecision and uncertainty. Depending on the application (databases, processes control, image processing…) knowledge representation and manipulation will involve the use of one qualitative operator or another.

CONCLUSION

In decision-making problems, the environment is often linguistic. For example, when evaluating the comfort of a room, linguistic terms expressing the "temperature" or the "sound level" such as "warm", "quiet"… are used. In this article, we have outlined a many-valued logic with a finite-valued system that suits well with these kinds of problems. Moreover we have presented several operators that deal with symbols and make computations on the linguistic symbols' indexes. In particular we focused on the four arithmetical operations (addition, subtraction, multiplication and division), on a modification operator and on two aggregators.

REFERENCES

Aït Younes, A., Truck, I., Akdag, H. (2007). Image Retrieval using Fuzzy Representation of Colors. *International Journal of Soft Computing – A Fusion of Foundations, Methodologies and Applications*, 11(3), 287-298.

Akdag, H., De Glas, M., & Pacholczyk, D. (1992). A Qualitative Theory of Uncertainty. *Fundamenta Informaticae*, 17(4), 333-362.

Akdag, H., Truck, I., Borgi, A., & Mellouli, N. (2001). Linguistic Modifiers in a Symbolic Framework. *Inter-*

national Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 9 (Supplement), 49-61.

Bouchon-Meunier, B., & Marsala, C. (2001). Linguistic modifiers and measures of similarity or resemblance. *9th IFSA World Congress*, Vancouver, 2195-2199.

Darwiche, A., & Ginsberg, M. (1992). A Symbolic Generalization of Probability Theory. *Proceedings of the Tenth National Conference of the American Association for Artificial Intelligence (AAAI)*, San Jose, California, 622-627.

Dubois, D., & Prade, H. (1985). A review of fuzzy set aggregation connectives. *Information Sciences*, 36, 85-121.

Gottwald, S. (2007). Many-Valued Logics. In *Handbook of the Philosophy of Sciences*, (5: Philosophy of Logic), 545-592.

Grabisch, M. (2001). On Preference Representation on an Ordinal Scale. *Proceedings of the 6th ECSQARU*, 18-28.

Herrera, F., Martínez, L., & Sánchez, P.J. (2005). Managing non-homogeneous information in group decision making. *European Journal of Operational Research*, 166(1), 115-132.

Khayata, M. Y., Pacholczyk, D., & Garcia, L. (2002). A Qualitative Approach to Syllogistic Reasoning. *Annals of Mathematics and Artificial Intelligence*, 34 (1-3), 131-159.

López de Mántaras, R., & Arcos, J. L. (2002). AI and Music: From Composition to Expressive Performances. *AI Magazine*, 23(3), 43-57.

Revault d'Allonnes, A., Akdag, H., & Poirel, O. (2007). Trust-moderated information-likelihood. A multi-valued logics approach. *Computation and Logic in the Real World, CiE 2007*, Siena, Italy.

Seridi, H., & Akdag, H. (2001). Approximate Reasoning for Processing Uncertainty. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 5(2), 108-116.

Truck, I., & Akdag, H. (2005). A Qualitative Approach for symbolic Data Manipulation under Uncertainty. In *Fuzzy Systems Engineering Theory and Practice, Series: Studies in Fuzziness and Soft Computing*, chapter 2. Springer.

Truck, I., & Akdag, H. (2006). Manipulation of Qualitative Degrees to Handle Uncertainty: Formal Models and Applications. *Knowledge and Information Systems*, 9 (4), 385-411.

Truck, I., Borgi, A., & Akdag, H. (2002). Generalized modifiers as an interval scale: Towards adaptive colorimetric alterations. *Proceedings of the 8th IBERAMIA*, 111-120.

Truck, I., Akdag, H., & Borgi, A. (2003). A Linguistic Approach of the Median Aggregator. *Proceedings of the 7th JCIS*, 147-150.

Zadeh, L. A. (2004). Precisiated Natural Language (PNL). *AI Magazine*, 25(3), 74-91.

KEY TERMS

Generalized Linguistic Modifiers: Linguistic Modifiers are tools that permit to express a more or less important modification on a linguistic value. The Generalized ones allow several kinds of modification (by dilation, erosion or conservation of the scales).

Linguistic Variables: Inside the framework called "Computing with Words", the variables used to manipulate human concepts are the *linguistic variables*.

Many-Valued Logic: A logic with more than two values to express the dependability of objects such as facts, sentences... The notion of partial membership underlies this logic.

Symbolic Aggregation: Permits to combine and summarize symbolic data. New aggregators must be defined because symbolic data can't assume the arithmetical properties of numerical values.

Symbolic Average: An original aggregator that combines symbolic data according to the algorithm of average computation. It also assumes the properties of the classical average.

Symbolic Data: Contains all kinds of data that are not intervals, neither fuzzy nor real numbers. In other words, it contains sentences, adverbs, symbols, words, sets of words, truth degrees, etc.

Symbolic Weighted Median: An aggregator that combines symbolic data according to the algorithm

of the weighted median, but with an original way to express the level of granularity (it uses the generalized symbolic modifiers). It also assumes the properties of the classical weighted median.

ENDNOTES

1 With M a non null positive integer.

2 $\mathcal{L}_M$ can also be considered as a *scale*.

3 A multiset is a generalization of a set: elements may *partially* belong to a multiset (to a certain degree).

4 They allow us to translate: $P(\varnothing) = 0$, $P(\Omega) = 1$, $P(\neg A) = 1 - P(A)$.

A User-Aware Multi-Agent System for Team Building

Pasquale De Meo
Università degli Studi Mediterranea di Reggio Calabria, Italy

Diego Plutino
Università Mediterranea di Reggio Calabria, Italy

Giovanni Quattrone
Università degli Studi Mediterranea di Reggio Calabria, Italy

Domenico Ursino
Università Mediterranea di Reggio Calabria, Italy

INTRODUCTION

In this chapter we present a system for the management of team building and team update activities in the current human resource management scenario. The proposed system presents three important characteristics that appear particularly relevant in this scenario. Firstly, it exploits a suitable standard to uniformly represent and handle expert skills. Secondly, it is highly distributed and, therefore, is well suited for the typical organization of the current job market where consulting firms are intermediating most job positions. Finally, it considers not only experts' technical skills but also their social and organizational capabilities, as well as the affinity degree possibly shown by them when they worked together in the past.

BACKGROUND

In the last years job market and organization have undergone deep changes. In fact, centralized job organization, where a company directly recruits its experts and assigns them to its activities, has been substituted by a distributed organization, where a company outsources most of its activities to external consulting firms. These last are often very large and complex; moreover, they frequently share the same clients or, even, the same projects and, consequently, they are enforced to cooperate. In the current human resource management scenario, most of available experts are recruited by these firms that send them to their final clients to run specific project tasks. It often happens that experts belonging to different consulting firms work together in the same project of interest to a final client (Meister, 1997).

In this highly distributed and flexible scenario, team building activities play a crucial role (Becerra & Fernandez, 2006). Interestingly enough, these activities are often known as task allocation activities (Dash, Vytelingum, Rogers, David & Jennings, 2007; Manisterski, David, Kraus & Jennings, 2006; Rahwan, Ramchurn, Dang, Giovannucci & Jennings, 2007). However, in the context of human resource management, team building problem presents some specific features (West, 2003) that make it more difficult and delicate to be handled w.r.t. the more general task allocation problem. Specifically, there are at least three main challenges to face.

The first challenge regards the highly distributed context; in fact, if experts of different consulting firms are enrolled to work together, then centralized team building approaches do not appear adequate.

The second challenge concerns the need of a standard for representing expert skills and capabilities (Hefke & Stojanovic, 2004; Harzallah, Leclere & Trichet, 2002; Biesalski, 2003); in fact, if consulting firms and/or their final clients use different ways to represent the skills of available experts and/or the skills desired for a project, then the comparison of experts and the matching between expert skills and project requirements may become difficult and unclear (Colucci, Di Noia, Di Sciascio, Donini & Ragone, 2007; Hefke & Stojanovic, 2004); as a consequence, there is a high

risk that constructed teams are not adequate for the projects assigned to them and, consequently, that final clients will be unsatisfied.

The third challenge regards the type of expert skills that must be considered during a team construction. In fact, technical skills, even if extremely important, cannot be the only criterion for choosing team members (Coutts & Gruman, 2005; Mehandjiev & Odgers, 1999). Indeed, other skills, such as social and organizational ones, appear equally important (Drach-Zahavy & Somech, 2002). As a confirmation of this claim, it is currently well known that a team wholly composed by technically talented experts is often characterized by a negative form of competition because its members tend to assert themselves each other by nature (LaFasto & Larson, 2001).

MAIN THRUST OF THE CHAPTER

System Overview

This chapter proposes a multi-agent system that exploits the Europass Curriculum Vitae (hereafter, Europass CV) standard to perform team building and team update activities. Europass CV is one of the standards defined by Europass (Europass, 2007); Europass is an initiative of the European Commission for achieving transparency of competencies and qualifications of an individual. Europass CV includes categories for describing past work experiences as well as technical skills and educational attainments; moreover, it records various additional competencies held by an individual, especially his social and organizational skills. The sections and the entries of this standard have been defined in such a way as to guarantee a company or an educational agency to obtain effective and suitable data about a candidate.

For each pair of experts, our system registers also the affinity degree showed by them in the projects where they worked together in the past. A project is represented as a collection of tasks; each task is described by a profile indicating the technical, social and organizational skills it requires to an expert if he wants to be eligible to execute it.

Each time a new team must be built for handling a project, our system generates a set of potentially adequate teams. Candidate teams are built on the basis of project tasks' requirements, as well as on the basis of technical, social and organizational skills of available experts. Once candidate teams have been built, our system selects, among them, that showing the highest value of internal cohesiveness. Our system is also capable of supporting team update activity; the algorithm underlying this task performs those updates that tend to optimize team performance and cohesiveness.

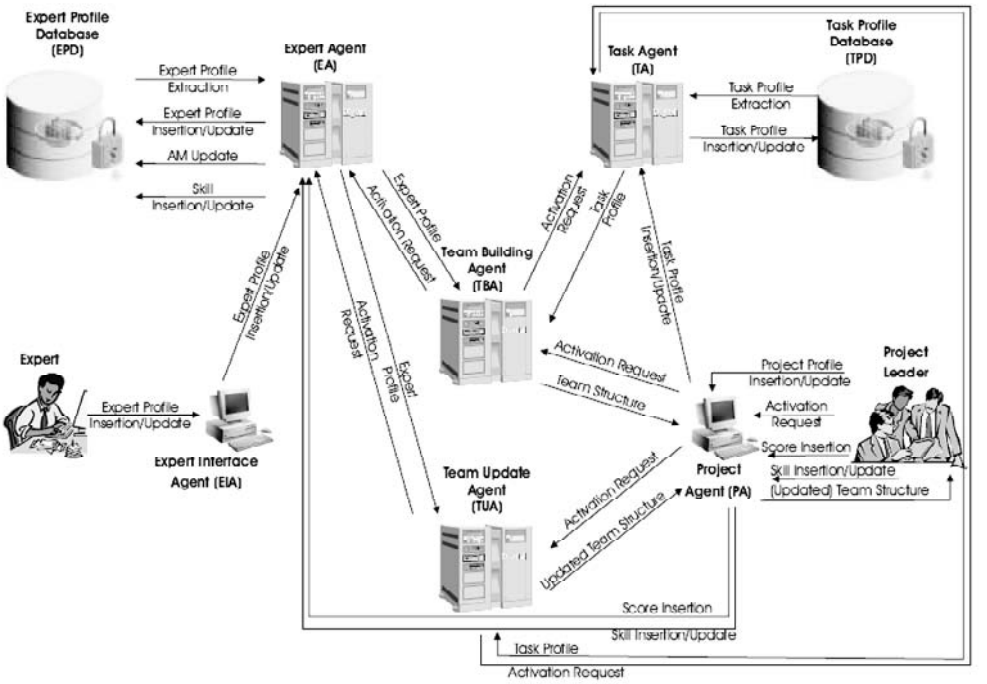
System Architecture and Behaviour

The architecture of our system is shown in Figure 1. It refers to a scenario in which *p* projects, each consisting of a certain number of tasks, must be handled by *n* experts (possibly enrolled from different consulting firms). Each project is uniquely assigned to a *project leader* who is in charge of insuring the fulfillment of its goals.

As shown in Figure 1, our system requires the presence of two databases, namely:

- *The Expert Profile Database* (hereafter, *EPD*); it stores both expert profiles and a support data structure called *Affinity Matrix* (hereafter, *AM*). The profile of an expert registers his personal data, the set of technical skills acquired by him in the past, as well as the set of his social and organizational skills. It is structured and organized according to the directives of the Europass CV standard. *AM* is a $n \times n$ matrix; its generic element *AM[i,l]* consists of a pair $<NP_{il}, S_{il}>$, where: *(i)* NP_{il} represents the overall number of projects on which the experts Exp_i and Exp_l worked together in the past; *(ii)* S_{il} stores the weighted sum of the scores obtained by Exp_i and Exp_l in the past; in other words, $S_{il} = \sum_{k=1}^{NP_{il}} k \cdot sc_{il}^{k}$ where sc_{il}^{k} is a score, belonging to the real interval [0,1], denoting how much Exp_i and Exp_l positively interacted in the fulfillment of the k^{th} project. The presence of *k* in this formula allows our system to give more importance to the scores obtained by Exp_i and Exp_l for their recent collaborations w.r.t. the scores assigned to them for their earliest collaborations.
- *The Task Profile Database* (hereafter, *TPD*); it stores the profiles of the tasks associated with the projects into consideration. The profile of a task registers the set of technical, social and organizational skills necessary to perform it.

Figure 1. Agents, databases, users and information flows involved in our system



The agents operating in our system are the following:

- *The Expert Interface Agent* (hereafter, *EIA*); it supports an expert in the insertion/update of his profile.
- *The Expert Agent* (hereafter, *EA*); it is in charge of handling all activities that someway concern the access to *EPD* data.
- *The Task Agent* (hereafter, *TA*); it supports all activities someway concerning the access to *TPD* data.
- *The Project Agent* (hereafter, *PA*); it supports all activities concerning the management of a project.
- *The Team Building Agent* (hereafter, *TBA*); it is the core of our system since it is in charge of building the best team for a new project. The team building approach performed by *TBA* receives a project consisting of a set $TaskSet = \{Task_1, \ldots, Task_m\}$ of m tasks, and a set $ExpSet = \{Exp_1, \ldots, Exp_n\}$ of n experts such that $m \leq n$. It returns a team represented as a set of m pairs of the form $\langle Exp_i, Task_j \rangle$, indicating that the expert Exp_i has been assigned to the task $Task_j$. The team building algorithm consists of three phases:
  - During the first phase a set *CandTeamSet* of q candidate teams is constructed. This activity is performed by considering how much the technical, social and organizational skills of each expert match with the corresponding requirements of the tasks of *TaskSet*.
  - During the second phase the teams of *CandTeamSet* are ranked according to their relevance to the tasks of *TaskSet*. Specifically, given a candidate team $Team_k = \{\langle Exp_{k1}, Task_{k1} \rangle, \ldots, \langle Exp_{km}, Task_{km} \rangle\}$, a *Team Adequacy* function, assessing the overall adequacy of a team in facing the tasks of *TaskSet*, is computed. This function is defined as follows:

$$TAdq(Team_k) = \frac{\sum_{u=1}^{m} Adq(Exp_{ku}, Task_{ku})}{m}$$

Here, *Adq* measures the adequacy of the expert Exp_{ku} for the task $Task_{ku}$; it is computed as a weighted sum of three functions measuring the adequacy of the technical, social and organizational skills of Exp_{ku} for the technical, social and organizational requirements of $Task_{ku}$. The values of *Adq*

and *TAdq* range in the real interval [0,1]; the higher the value of Team Adequacy is, the more adequate $Team_k$ will be for *TaskSet*. The teams of *CandTeamSet* are ranked according to the corresponding Team Adequacy values in a descending order. After this ranking activity, the top *h* teams are selected and put in a set *TopCandSet*.

- During the third phase a suitable *Cohesiveness* function is computed for each element of *TopCandSet*. The cohesiveness of a team takes the affinities of the corresponding members into account as stored in the Affinity Matrix. After this, the team having the maximum cohesiveness value is selected; if two or more teams have the same maximum cohesiveness value, then one of the teams having the maximum Team Adequacy value is randomly selected among them. It is worth pointing out that the team selected at the end of this phase is not necessarily that having the highest Team Adequacy value, among those of *TopCandSet*. This choice allows the third challenge specified in the Background to be faced.

- *The Team Update Agent* (hereafter, *TUA*); it is in charge of incrementally updating a team for a project. It is activated by a project leader, via the associated *PA*, when it is necessary to carry out some modification on an already existing team. However, it can also autonomously suggest a team update when it has found that a new expert is available for a project and that he can improve the corresponding Team Adequacy without lowering the corresponding cohesiveness. Our Team Update algorithm implements a hill climbing meta-heuristics. Specifically, it constructs a new candidate team by applying on the current team a suitable operation aiming to produce the highest increase of *TAdq* without lowering the Cohesiveness function. In order to preserve the network of human relationships constructed previously, the implemented meta-heuristics must satisfy the following requirements: *(i)* the structure of a team should be preserved as much as possible; *(ii)* a new free expert should not privileged w.r.t. already existing team members.

In order to describe the behaviour of our system, two possible operating scenarios, namely the construction of a new team and the incremental update of an existing one, must be considered.

In the first scenario the behaviour of our system is as follows: at the beginning, a project leader activates the corresponding *PA* and provides information about the new project and the corresponding tasks. *PA* sends task information to *TA* which is in charge of storing it in *TPD*. After this, *PA* activates *TBA* to construct a team for the new project. *TBA* first activates *EA* and requires it the profile of the experts who might be involved in the project; then it activates *TA* and requires the profile of the tasks of the current project. After this, it applies the team building algorithm described above and constructs the team for the project. Finally, it sends its results to the project leader via *PA*. At the end of the project execution, the project leader is required to fill an appraisal form in order to assign an affinity score to each pair of team members. These scores are used to update the Affinity Matrix *AM* stored in *EPD*. Once *AM* has been updated, the project leader must fill a further form to specify, for each expert, the skills acquired or improved by him during the project execution. This information is stored in *EPD* via *PA* and *EA*.

As far as the second reference scenario is concerned, three possible situations might occur, namely: *(i)* an expert leaves a team; *(ii)* a new expert is available and his skills are particularly adequate for one or more project tasks; *(iii)* one or more tasks must be added to the current project. The first and the third situations make the current team obsolete; in these cases *TUA* is activated by the project leader; its behaviour is, therefore, reactive. By contrast, the second situation does not necessarily impose a team update; in fact, it requires to verify the advantages and the disadvantages arising if the new expert would be added to the team; in this situation *TUA* shows a proactive behaviour since it autonomously activates the verification specified above when it recognizes the availability of a new expert. Observe that this team update problem might be seen as a special case of the team building problem examined above; as a consequence, it would be possible to determine a new team configuration by ignoring the existing one. However, in a real scenario, when a team is updated, managers try to minimize the modifications on task assignments; in fact, modifications are performed only if they are strictly necessary or if their advantages significantly exceed their disadvantages.

This reasoning led us to define suitable policies for incrementally updating a team.

Each time a team must be incrementally updated, our system behaves as follows. First *TUA* activates *TA* and requires information about the tasks of the project into consideration; after this, it activates *EA* and requires information about both those experts already assigned to these tasks and those experts currently eligible to be aggregated to the team. Once all profiles are available, *TUA* performs the incremental team update activity. Finally, it sends obtained results to the project leader via *PA*.

Discussion

Our system is capable of facing all the three challenges mentioned in the Background section. First, in order to determine the adequacy of an expert for a task, it examines not only the technical skills of the expert and the technical requirements of the task, but also the corresponding social and organizational skills and requirements. In addition, it does not necessarily select the team showing the highest technical adequacy with the project into examination; by contrast, it selects, among those teams which satisfy the project technical minimal requirements, that showing the highest internal cohesiveness. This behaviour allows our system to face the third challenge mentioned in the Background section.

The exploitation of Europass CV standard favours a uniform representation of both the skills owned by experts and those required by project tasks. Interestingly enough, thanks to the choice of using this standard, skill representation adopted by our system is derived from real life, i.e., from the real format used by experts to represent their curricula vitae. This choice allows our system to face the second challenge mentioned in the Background section.

Finally, multi-agent paradigm is well suited to manage the largely distributed scenario characterizing the current job market. Moreover, the presence of an agent that registers the profiles of available experts and updates them at the end of a project, on the basis of their behaviour, allows our system to fruitfully benefit from the learning capability typical of Intelligent Agents. Finally, message exchanges designed for our system allow experts and project leaders to easily communicate, on the basis of a standard "language" (i.e., the Europass CV standard), in such a way that the former can precisely specify their skills and capabilities to the latter. Thus, our system is capable of facing also the first challenge mentioned in the Background section.

FUTURE TRENDS

As for future trends, a team building system could be integrated in a more complex Enterprise Resource Planning system capable of fully bringing out the human capital of a firm. Specifically, it would be possible to design tools for identifying and tracking "high-potential" employees; this would support firm managers to implement career advancement plans and training courses in such a way as to prepare talented junior experts to assume leadership roles.

In addition, it is possible to design a performance-oriented compensation process, i.e., an incentive mechanism that stimulates employees to achieve ambitious business goals. To this purpose, a team building system could be integrated with sophisticated machine learning techniques, like the Q-Learning algorithm.

CONCLUSION

In this chapter we have proposed a new system aiming to handle team building and team update activities in the current job market scenario. The proposed system exploits the Europass CV standard to uniformly represent and handle expert skills. Moreover, it is highly distributed and, therefore, is particularly adequate to the typical organization of the current job market where consulting firms are currently intermediating most job positions. Finally, it considers not only technical skills of experts but also their social and organizational capabilities, as well as the affinity showed by them in the past.

REFERENCES

Becerra-Fernandez, I. (2006). Searching for experts on the Web: A review of contemporary expertise locator systems. *ACM Transactions on Internet Technology*, *6*(4), 333-355.

Biesalski, E. (2003). Knowledge management and e-human resource management. *Proc. of the International Workshop on Knowledge and Experience Management* (FGWM 2003), pp. 167-172, Karlsruhe, Germany.

Colucci, S., Di Noia, T., Di Sciascio, E., Donini, F.M. & Ragone, A. (2007). Semantic-based skill management for automated task assignment and courseware composition. *Journal of Universal Computer Science*. Forthcoming.

Coutts., L.M. & Gruman, J. (2005). Social psychology applied to organizations. In *Applied Social Psychology - Understanding and Addressing Social and Practical Problems*. SAGE Publications, Inc.

Dash, R. K., Vytelingum, P., Rogers, A., David, E. & Jennings, N. (2007) Market-based task allocation mechanisms for limited capacity suppliers. *IEEE Transactions on Systems, Man, and Cybernetics - Part A, 37*(3), 391-405.

Drach-Zahavy, A. & Somech, A. (2002). Team heterogeneity and its relationship with team support and team effectiveness. *Journal of Educational Administration, 40*(1), 44-66.

Europass. From http://europass.cedefop.europa.eu/europass/home/hornav/Introduction/navigate.action

Harzallah, M., Leclere, M. & Trichet, F. (2002). CommOnCV: Modelling the competencies underlying a curriculum vitae. *Proc. of the International Conference on Software Engineering and Knowledge Engineering* (*SEKE 2002*), pp. 65-71, Ischia, Italy. ACM Press.

Hefke, M. & Stojanovic, L. (2004). An ontology-based approach for competence bundling and composition of ad-hoc teams in an organisation. *Proc. of the International Conference on Knowledge Management* (I-KNOW 2004), pp. 126-134, Graz, Austria.

LaFasto, F.M.J. & Larson., C. (2001). *When teams work best*. Sage Publications, Inc.

Manisterski, E., David, E., Kraus, S. & Jennings N.R. (2006). Forming efficient agent groups for completing complex tasks. *Proc. of the ACM International Joint Conference on Autonomous Agents and Multiagent Systems* (AAMAS 2006), pp. 834-841, Hakodate, Japan, ACM Press

Mehandjiev, N. & Odgers, B. (1999). SAMBA: Agent-supported visual interactive control for distributed team building and empowerment. *BT Technology Journal, 17*(4), 72-77.

Meister, D.H. (1997). *Managing the professional service firm*. New York: Free Press.

Rahwan, T., Ramchurn, S.D., Dang, V.D., Giovannucci, A. & Jennings, R. (2007) Anytime optimal coalition structure generation. *Proc. of the AAAI Conference on Artificial Intelligence* (AAAI 2007), Vancouver, Canada, 1184-1190, AAAI Press.

West, M.A. (2003). *Effective teamwork*. Oxford, UK: Blackwell Publishing.

KEY TERMS

Agent: A computational entity capable of both perceiving dynamic changes in the environment it is operating in and autonomously performing user delegated tasks, possibly by communicating and co-operating with other similar entities.

Europass: An initiative of the European Commission that aims to provide people with tools and services capable of making their skills and competences clearly understood in different countries.

Europass Curriculum Vitae: One of the standards proposed by Europass. It allows the learning and the work experiences of an individual to be specified; the section and the entries of this standard have been defined in such a way as to guarantee a company or an educational agency to obtain effective and suitable data about a candidate.

Multi-Agent System (MAS): A loosely coupled network of software agents that interact to solve problems that are beyond the individual capacities or knowledge of each of them.

Team Building: The process of constructing a team capable of handling a project consisting of one or more tasks.

Team Update: The process of updating an already constructed team.

User Modelling: The process of gathering information specific to each user either explicitly or implicitly.

User Profile: A model of a user representing both her/his preferences and her/his behaviour.

Using Dempster–Shafer Theory in Data Mining

Malcolm J. Beynon
Cardiff University, UK

INTRODUCTION

The origins of Dempster-Shafer theory (DST) go back to the work by Dempster (1967) who developed a system of upper and lower probabilities. Following this, his student Shafer (1976), in their book "A Mathematical Theory of Evidence" developed Dempster's work, including a more thorough explanation of belief functions, a more general term for DST. In summary, it is a methodology for evidential reasoning, manipulating uncertainty and capable of representing partial knowledge (Haenni & Lehmann, 2002; Kulasekere, Premaratne, Dewasurendra, Shyu, & Bauer, 2004; Scotney & McClean, 2003).

The perception of DST as a generalisation of Bayesian theory (Shafer & Pearl, 1990), identifies its subjective view, simply, the probability of an event indicates the degree to which someone believes it. This is in contrast to the alternative frequentist view, understood through the "Principle of I sufficient reasoning", whereby in a situation of ignorance a Bayesian approach is forced to evenly allocate subjective (additive) probabilities over the frame of discernment. See Cobb and Shenoy (2003) for a contemporary comparison between Bayesian and belief function reasoning.

The development of DST includes analogies to rough set theory (Wu, Leung, & Zhang, 2002) and its operation within neural and fuzzy environments (Binaghi, Gallo, & Madella, 2000; Yang, Chen, & Wu, 2003). Techniques based around belief decision trees (Elouedi, Mellouli, & Smets, 2001), multi-criteria decision making (Beynon, 2002) and non-paramnteric regression (Petit-Renaud & Denœux, 2004), utilise DST to allow analysis in the presence of uncertainty and imprecision. This is demonstrated, in this article, with the 'Classification and Ranking belief Simplex' (CaRBS) technique for object classification, see Beynon (2005a).

BACKGROUND

The terminology inherent with DST starts with a finite set of hypotheses Θ (the frame of discernment). A *basic probability assignment* (*bpa*) or mass value is a function m: $2^\Theta \rightarrow [0, 1]$ such that $m(\varnothing) = 0$ ($\varnothing$ - the empty set) and

$$\sum_{A \in 2^\Theta} m(A) = 1$$ (2^Θ - the power set of Θ).

If the assignment $m(\varnothing) = 0$ is not imposed then the transferable belief model can be adopted (Elouedi, Mellouli, & Smets, 2001; Petit-Renaud & Denœux, 2004). Any $A \in 2^\Theta$, for which $m(A)$ is non-zero, is called a focal element and represents the exact belief in the proposition depicted by A. From a single piece of evidence, a set of focal elements and their mass values can be defined a *body of evidence* (BOE).

Based on a BOE, a *belief* measure is a function *Bel*: $2^\Theta \rightarrow [0, 1]$, defined by,

$$Bel(A) = \sum_{A \subseteq B} m(B),$$

for all $A \subseteq \Theta$. It represents the confidence that a specific proposition lies in A or any subset of A. The *plausibility* measure is a function *Pls*: $2^\Theta \rightarrow [0, 1]$, defined by,

$$Pls(A) = \sum_{A \cap B \neq \varnothing} m(B),$$

for all $A \subseteq \Theta$. Clearly $Pls(A)$ represents the extent to which we fail to disbelieve A. these measures are directly related to one another, $Bel(A) = 1 - Pls(\neg A)$ and $Pls(A) = 1 - Bel(\neg A)$, where $\neg A$ refers to its compliment 'not A'.

To collate two or more sources of evidence (e.g. $m_1(\cdot)$ and $m_2(\cdot)$), DST provides a method to combine them, using Dempster's rule of combination. If $m_1(\cdot)$ and $m_2(\cdot)$ are two independent BOEs, then the function $(m_1 \oplus m_2)$: $2^\Theta \rightarrow [0, 1]$, defined by:

$$(m_1 \oplus m_2)(y) = \begin{cases} 0 & y = \varnothing \\ (1-\kappa)^{-1} \sum_{A \cap B = y} m_1(A) m_2(B) & y \neq \varnothing \end{cases}$$

where $\kappa = \sum_{A \cap B = \varnothing} m_1(A) m_2(B)$,

is a mass value with $y \subseteq \Theta$. The term $(1 - \kappa)$, can be interpreted as a measure of conflict between sources. It is important to take this value into account for evaluating the quality of combination: when it is high, the combination may not make sense and possible lead to questionable decisions (Murphy, 2000). One solution to mitigate conflict is to assign noticeable levels of ignorance to all evidence, pertinently the case when low level measurements are considered (Gerig, Welti, Guttman, Colchester, & Szekely, 2000).

To demonstrate the utilization of DST, the example of the murder of Mr. Jones is considered, where the murderer was one of three assassins, Peter, Paul and Mary, so the frame of discernment Θ = {Peter, Paul, Mary}. There are two witnesses. Witness 1, is 80% sure that it was a man, the concomitant BOE, defined $m_1(\cdot)$, includes $m_1(\{\text{Peter, Paul}\}) = 0.8$. Since we know nothing about the remaining mass value it is considered ignorance and allocated to Θ, hence $m_1(\{\text{Peter, Paul, Mary}\}) = 0.2$. Witness 2, is 60% confident that Peter was leaving on a jet plane when the murder occurred, so a BOE defined $m_2(\cdot)$ includes, $m_2(\{\text{Paul, Mary}\}) = 0.6$ and $m_2(\{\text{Peter, Paul, Mary}\}) = 0.4$.

The aggregation of these two sources of information (evidence), using Dempster's combination rule, is based on the intersection and multiplication of focal elements and mass values from the BOEs, $m_1(\cdot)$ and $m_2(\cdot)$. Defining this BOE $m_3(\cdot)$, it can be found; $m_3(\{\text{Paul}\}) = 0.48$, $m_3(\{\text{Peter, Paul}\}) = 0.32$, $m_3(\{\text{Paul, Mary}\}) = 0.12$ and $m_3(\{\text{Peter, Paul, Mary}\}) = 0.08$. This combined evidence has a more spread-out allocation of mass values to varying subsets of the frame of discernment Θ. Further, there is a general reduction in the level of ignorance associated with the combined evidence. In the case of the belief (*Bel*) and plausibility (*Pls*) measures, considering the subset {Peter, Paul}, then $Bel_3(\{\text{Peter, Paul}\}) = 0.8$ and $Pls_3(\{\text{Peter, Paul}\}) = 1.0$. Smets (1990) offers a comparison on a variation of this example with how it would be modelled using traditional probability and the transferable belief model.

A second larger example supposes that the weather in New York at noon tomorrow is to be predicted from the weather today. We assume that it is in exactly one of the three states: dry (D), raining (R) or snowing (S). Hence the frame of discernment is represented by Θ = {D, R, S}. Let us assume that two pieces of evidence have been gathered: *i*) The temperature today is below freezing, and *ii*) The barometric pressure is falling; i.e., a storm is likely. These pieces of evidence are represented by the two BOE, $m_{\text{freeze}}(\cdot)$ and $m_{\text{storm}}(\cdot)$, respectively, and are reported in Table 1.

For each BOE in Table 1, the exact belief (mass) is distributed among the focal elements (excluding ∅). For $m_{\text{freeze}}(\cdot)$, greater mass is assigned to {S} and {R, S}, for $m_{\text{storm}}(\cdot)$, greater mass is assigned to {R} and {R, S}. Assuming that $m_{\text{freeze}}(\cdot)$ and $m_{\text{storm}}(\cdot)$ represent items of evidence which are independent of one another, a new BOE $m_{\text{both}}(\cdot)$ is given by Dempster's rule of combination; with $m_{\text{both}}(\cdot) = m_{\text{freeze}}(\cdot) \oplus m_{\text{storm}}(\cdot)$, shown in Table 2.

The BOE $m_{both}(\cdot)$ represented in Table 2 has a lower level of local ignorance ($m_{both}(\Theta) = 0.0256$), than both of the original BOEs, $m_{\text{freeze}}(\cdot)$ and $m_{\text{storm}}(\cdot)$. Amongst the other focal elements, more mass is assigned to {R} and {S}, a consequence of the greater mass assigned to the associated focal elements in the two constituent

Table 1. Mass values and focal elements for $m_{freeze}(\cdot)$ and $m_{storm}(\cdot)$

BOE	∅	{D}	{R}	{S}	{D, R}	{D, S}	{R, S}	Θ
$m_{\text{freeze}}(\cdot)$	0.0	0.1	0.1	0.2	0.1	0.1	0.2	0.2
$m_{\text{storm}}(\cdot)$	0.0	0.1	0.2	0.1	0.1	0.1	0.3	0.1

Table 2. Mass values and focal elements for the BOE $m_{both}(\cdot)$

BOE	∅	{D}	{R}	{S}	{D, R}	{D, S}	{R, S}	Θ
$m_{\text{freeze}}(\cdot)$	0.0	0.1282	0.2820	0.2820	0.0513	0.0513	0.1795	0.0256

BOEs. The other focal elements all exhibit net losses in their mass values. As with the assassin example, measures of belief (*Bel*) and plausibility (*Pls*) can be found, to offer evidence (confidence) on combinations of states representing tomorrow's predicted weather. For example, the degree of belief in the proposition {R, S} based on the combined evidence, $Bel_{both}(\{R, S\}) = 0.2820 + 0.2820 + 0.1795 = 0.7435$, is substantially higher than that based on $m_{freeze}(\cdot)$ ($Bel_{freeze}(\{R, S\}) = 0.5$) and $m_{storm}(\cdot)$ ($Bel_{storm}(\{R, S\}) = 0.6$).

This section is closed with some cautionary words still true to this day (Pearl, 1990), "Some people qualify any model that uses belief functions as Dempster-Shafer. This might be acceptable provided they did not blindly accept the applicability of both Dempster's rule of combination. Such critical - and in fact often inappropriate - use of such rules explains most of the errors encountered in the so called Dempster-Shafer literature."

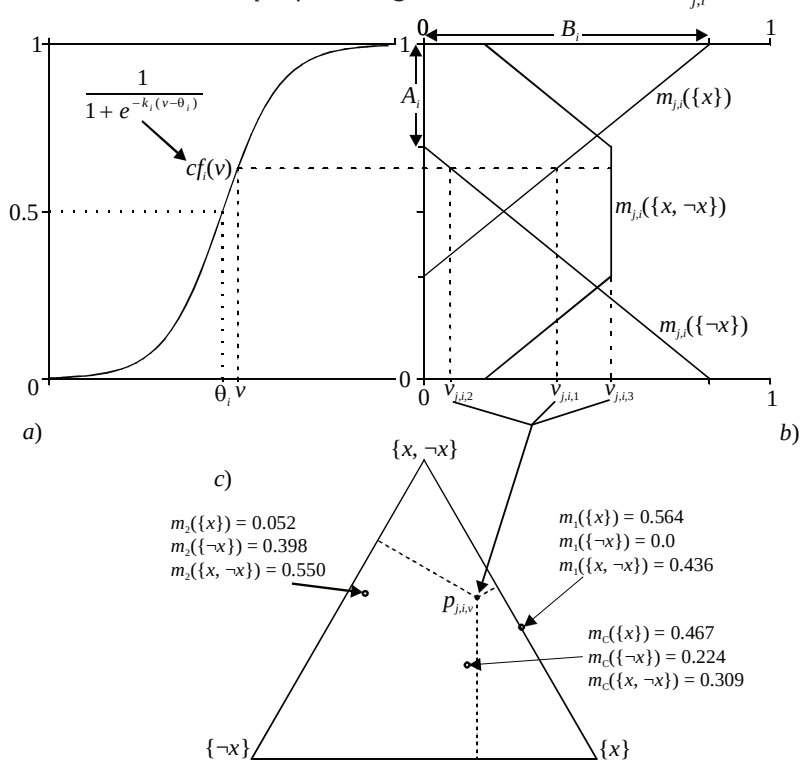
MAIN THRUST

This section outlines one of many different techniques which utilizes DST, called CaRBS (Classification and Ranking Belief System), within a data mining environment. It is a data mining technique for the classification, and subsequent prediction, of objects to a given hypothesis (x) and its compliment ($\neg x$), using a series of characteristic values (Beynon, 2005a). Since its introduction it has been applied in the areas of, credit ratings (Beynon, 2005b), osteoarthritic knee analysis (Jones, Beynon, Holt, & Roy, 2006) and US state long term care systems (Beynon & Kitchener, 2005).

The rudiments of the CaRBS technique are based on DST, and with two exhaustive outcomes works on a binary frame of discernment (BFOD). Subsequently, the aim of the CaRBS is to construct a BOE ($m(\{x\})$) for each characteristic value to quantify its evidential support for the classification of an object to x ($m(\{x\})$), $\neg x$ ($m(\{\neg x\})$) and concomitant ignorance $\{x, \neg x\}$ ($m(\{x, \neg x\})$), see Figure 1 for a brief exposition of its operational stages.

In Figure 1, stage *a*) shows the transformation of a characteristic value $v_{j,i}$ (j^{th} object, i^{th} characteristic) into a confidence value $cf_i(v_{j,i})$, using a sigmoid function, defined with the control variables k_i and θ_i. Stage *b*) transforms a characteristic value's confidence value $cf_i(v_{j,i})$ into a characteristic BOE, defined $m_{j,i}(\cdot)$, made

Figure 1. Stages within the CaRBS technique for a single characteristic value $v_{j,i}$



up of the three mass values, $m_{j,i}(\{x\})$, $m_{j,i}(\{\neg x\})$ and $m_{j,i}(\{x, \neg x\})$, from Gerig, Welti, Guttman, Colchester, & Szekely (2000), they are defined by;

$$m_{j,i}(\{x\}) = \frac{B_i}{1-A_i} cf_i(v_{j,i}) - \frac{A_i B_i}{1-A_i}, \; m_{j,i}(\{\neg x\}) = \frac{-B_i}{1-A_i} cf_i(v_{j,i}) + B_i,$$

and $m_{j,i}(\{x, \neg x\}) = 1 - m_{j,i}(\{x\}) - m_{j,i}(\{\neg x\})$,

where A_i and B_i are two further control variables utilised. When either $m_{j,i}(\{x\})$ or $m_{j,i}(\{\neg x\})$ are negative they are set to zero (before the calculation of $m_{j,i}(\{x, \neg x\})$). The control variable A_i depicts the dependence to $m_{j,i}(\{x\})$ on $cf_i(v_{j,i})$ and B_i is the maximum value assigned to $m_{j,i}(\{x\})$ or $m_{j,i}(\{\neg x\})$. Stage *c*) shows the mass values in a BOE $m_{j,i}(\cdot)$; $m_{j,i}(\{x\})$, $m_{j,i}(\{\neg x\})$ and $m_{j,i}(\{x, \neg x\})$, can be represented as a simplex coordinate (single point - $p_{j,i,v}$) in a simplex plot (equilateral triangle). That is, a point $p_{j,i,v}$ exists within an equilateral triangle such that the least distance from $p_{j,i,v}$ to each of the sides of the equilateral triangle are in the same proportion (ratio) to the values $v_{j,i,1}$, $v_{j,i,2}$ and $v_{j,i,3}$. Each corner (vertex) of the equilateral triangle is labelled with one of the three focal elements in the BOE (in the case of a BFOD – as here).

When the characteristics c_i, $i = 1, \ldots, n_C$, describe an object o_j, $j = 1, \ldots, n_O$, individual characteristic BOEs are constructed. Dempster's rule of combination is used to combine these (independent) characteristic BOEs to produce an object BOE $m_j(\cdot)$, associated with an object o_j and its level of classification to x or $\neg x$. To illustrate the method of combination employed here, two example BOEs, $m_1(\cdot)$ and $m_2(\cdot)$ are considered, with mass values in each BOE given in the vector form $[m_j(\{x\}), m_j(\{\neg x\}), m_j(\{x, \neg x\})]$ as, [0.564, 0.000, 0.436] and [0.052, 0.398, 0.550], respectively. The combination of $m_1(\cdot)$ and $m_2(\cdot)$ is evaluated to be [0.467, 0.224, 0.309], further illustrated in Figure 1*c*, where the simplex coordinates of the BOEs $m_1(\cdot)$ and $m_2(\cdot)$ are shown along with that of the resultant combined BOE $m_C(\cdot)$. Inspection of the simplex coordinate representation of the BOEs shows the effect of the combination of $m_1(\cdot)$ and $m_2(\cdot)$, including a decrease in the ignorance associated with the resultant BOE $m_C(\cdot)$.

The effectiveness of the CaRBS technique is governed by the values assigned to the incumbent control variables, k_i, θ_i, A_i and B_i, $i = 1, \ldots, n$. When the classification of each object is known, the necessary configuration is considered as a constrained optimisation problem. Since these variables are continuous in nature the recently introduced trigonometric differential evolution (TDE) method is utilised (Storn and Price, 1997; Fan and Lampinen, 2003). When the classification of a number of objects to some hypothesis and its complement is known, the effectiveness of a configured CaRBS system can be measured by a defined objective function (OB), two of which are described here.

Beynon (2005a) included an objective function which maximised the certainty in each object's final classification (minimising ambiguity and ignorance). This first OB, defined OB1, uses the mean simplex coordinates of the final object BOEs of objects known to be classified to $\{x\}$ or $\{\neg x\}$, the sets of objects are termed equivalence classes ($E(\cdot)$), defined $E(x)$ and $E(\neg x)$. Then the mean simplex coordinate of an equivalence class $E(\cdot)$ is

$$\left(\frac{1}{|E(\cdot)|} \sum_{o_j \in E(\cdot)} x_j, \frac{1}{|E(\cdot)|} \sum_{o_j \in E(\cdot)} y_j \right),$$

where (x_j, y_j) is the simplex coordinate of the object BOE associated with the object o_j and $|E(\cdot)|$ is the number of objects in the respective equivalence class. An OB traditionally uses a best fitness of zero, it follows the OB1 is seen in Box 1. Where (x_H, y_H) and (x_N, y_N) are the simplex coordinates of the $\{x\}$ and $\{\neg x\}$ vertices in the domain of the simplex plot, respectively. In general, the OB1 has range $0 \leq OB1 \leq \sqrt{(x_N - x_H)^2 + (y_N - y_H)^2}$. For a simplex plot with vertex coordinates (1, 0), (0, 0) and $(0.5, \sqrt{3}/2)$, it is an equilateral triangle with unit side, then $0 \leq OB1 \leq 1$.

The second objective function considered here was constructed in Beynon (2005b), which in contrast to OB1, considers optimisation through the minimisation of only the ambiguity in the classification of the set of objects. For objects in the equivalence classes, $E(x)$ and $E(\neg x)$, the optimum solution is to maximise the difference values $(m_j(\{x\}) - m_j(\{\neg x\}))$ and $(m_j(\{\neg x\}) - m_j(\{x\}))$, respectively. This objective function, defined OB2, where optimisation is minimisation with a lower limit of zero, is given by:

Box 1.

$$\mathrm{OB1}=\frac{1}{2}\left(\sqrt{\left(x_{\mathrm{H}}-\frac{1}{|E(x)|}\sum_{o_j\in E(x)}x_j\right)^2+\left(y_{\mathrm{H}}-\frac{1}{|E(x)|}\sum_{o_j\in E(x)}y_j\right)^2}+\sqrt{\left(x_{\mathrm{N}}-\frac{1}{|E(\neg x)|}\sum_{o_j\in E(\neg x)}x_j\right)^2+\left(y_{\mathrm{N}}-\frac{1}{|E(\neg x)|}\sum_{o_j\in E(\neg x)}y_j\right)^2}\right),$$

$$\mathrm{OB2}=\frac{1}{4}\left(\begin{array}{l}\dfrac{1}{|E(x)|}\displaystyle\sum_{o_j\in E(x)}(1-m_j(\{x\})+m_j(\{\neg x\}))+\\ \dfrac{1}{|E(\neg x)|}\displaystyle\sum_{o_j\in E(\neg x)}(1+m_j(\{x\})-m_j(\{\neg x\}))\end{array}\right).$$

In the limit, each of the difference values, $(m_j(\{x\}) - m_j(\{\neg x\}))$ and $(m_j(\{\neg x\}) - m_j(\{x\}))$, has domain [–1, 1], then $0 \le \mathrm{OB2} \le 1$. It is noted, maximising a difference value such as $(m_j(\{x\}) - m_j(\{\neg x\}))$ only indirectly affects the associated ignorance $(m_j(\{x, \neg x\}))$, rather than making it a direct issue, as with OB1.

To demonstrate the CaRBS technique and the defined objective functions, a small example data set is considered, made up of only ten objects, described by three characteristic values and whether they are associated with x or $\neg x$, their details given in Table 3.

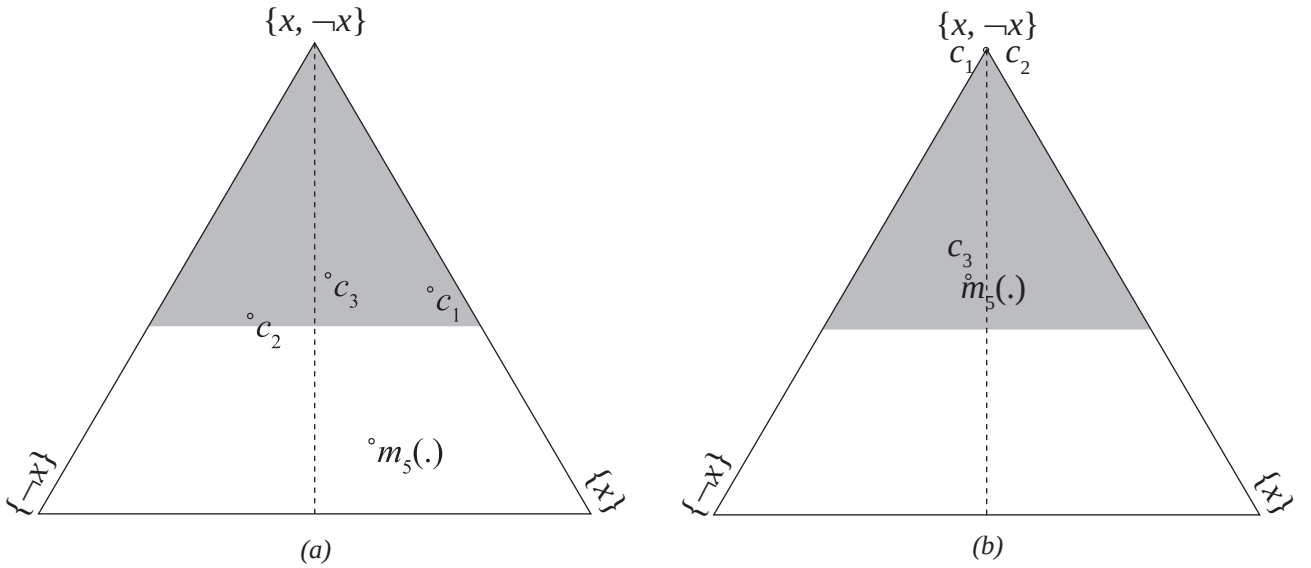
Using the CaRBS technique, to assimilate ignorance in the characteristics (their evidence), an upper bound of 0.6 is placed on each of the B_i control variables. The data set was also standardized (zero mean and unit standard deviation) allowing bounds on the k_i and θ_i control variables of [–3, 3] and [–2, 2], respectively (limited marginal effects on results). To utilize TDE, parameters are required for its operation. Following Fan and Lampinen (2003), amplification control F = 0.99, crossover constant CR = 0.85, trigonometric mutation probability Mt = 0.05 and number of parameter vectors NP = 10 × number of control variables = 90. The TDE was then applied and the subsequent control variables used to construct characteristics BOEs, and post combination an object BOE for each object, see Figure 3.

The first two simplex plots in Figure 3 represent the classification of the object o_5, depending on whether the objective functions, OB1 and OB2, were employed. In each simplex plot, the grey shaded region shows the domain that the characteristic BOEs can exist in. In Figure 3*a*, using OB1, all three characteristics offer evidence to its classification, two of which are offering incorrect evidence (c_1 and c_3), since the object o_5 is known to be associated with $\neg x$, see Table 3. These characteristic BOEs produce an object BOE $m_5(\cdot)$ shown further down, which is nearest the $\{x\}$ vertex, hence incorrectly classifying this object. In Figure 3*b*, using OB2, only c_3 offers evidence, the other two only offer total ignorance, so at the $\{x, \neg x\}$ vertex. The final object BOE is the same as for c_3, but is to the left hand-side

Table 3. Details of example data set

	o_1	o_2	o_3	o_4	o_5	o_6	o_7	o_8	o_9	o_{10}
c_1	0.61	1.00	1.12	1.50	0.62	1.27	0.73	0.51	0.45	0.58
c_2	0.98	0.75	0.68	0.46	0.80	0.64	1.18	0.95	0.85	0.71
c_3	0.02	0.15	0.31	–0.12	0.10	0.03	–0.02	0.11	–0.06	0.03
d_1	$\neg x$	$\neg x$	$\neg x$	$\neg x$	$\neg x$	x	x	x	x	x

Figure 3. Simplex plot based classification details of object o_5 (3a and 3b) and all objects (3c and 3d) using the objective functions OB1 (3a and 3c) and OB2 (3b and 3d)



of the vertical dashed line so is classified to being $\neg x$, which is now a correct classification.

The results in Figures 3*c* and 3*d* show the effect of the different objective functions utilized in the final classification of all ten objects (labels, $\neg x$ - circle and x - cross). While there is more ignorance associated with the results in Figure 3*d*, due to the allowance for ignorance in results (using OB2). While only a small data set, the results to show how the presence of ignorance is an importance issue and may effect the more traditional results of classification accuracy often used.

FUTURE TRENDS

The practical utilization and understanding of DST is a continuing debate, specific to its inherent generality, including the notion of ignorance (partial knowledge). This includes the practical construction of the required BOEs, without the constraining influence of an expert. This is an ardent possibility with the more subjective nature of DST over the relative frequentist form with a Bayesian based approach.

The CaRBS technique discussed is a data mining technique particular to object classification (and ranking). The trigonometric differential evolution method of constrained optimisation mitigates the influence of expert opinion in assigning values to the incumbent control variables. However, the effect of the limiting bounds on these control variables still needs further understanding.

CONCLUSION

The utilization of DST highlights the acknowledgement of the attempt to incorporate partial knowledge (ignorance/uncertainty) in functional data mining. The understanding and formulization of what ignorance is, and how to quantify it in general and application specific studies is an underlying problem. The future will undoubtedly produce directions of appropriate (and inappropriate) definitions to its quantification. Indeed, the results from the different objective functions considered to quantify the configuration undertaken show that allowing or not allowing for the presence of ignorance can affect results.

REFERENCES

Beynon, M. (2002). DS/AHP Method: A Mathematical Analysis, including an Understanding of Uncertainty,

European Journal of Operational Research, *140*(1), 149-165.

Beynon, M. J. (2005a). A Novel Technique of Object Ranking and Classification under Ignorance: An Application to the Corporate Failure Risk Problem. *European Journal of Operational Research*, *167*(2), 493-517.

Beynon, M. J. (2005b). A Novel Approach to the Credit Rating Problem: Object Classification Under Ignorance. *International Journal of Intelligent Systems in Accounting, Finance and Management*, *13*, 113-130.

Beynon, M. J., & Kitchener, M. (2005). Ranking the 'Balance' of State Long-Term Care Systems: A Comparative Exposition of the SMARTER and CaRBS Techniques. *Health Care Management Science*, *8*, 157-166.

Binaghi, E., Gallo, I., & Madella, P. (2000). A neural model for fuzzy Dempster-Shafer theory classifiers. *International Journal of Approximate Reasoning*, *25*, 89-121.

Cobb, B. R., & Shenoy, P. P. (2003). A Comparison of Bayesian and Belief Function Reasoning. *Information Systems Frontiers*, 5(4), 345-358.

Dempster, A. P. (1967). Upper and lower probabilities induced by a multiple valued mapping. *Ann. Math. Statistics*, *38*, 325-339.

Elouedi, Z., Mellouli, K., & Smets, P. (2001). Belief decision trees: Theoretical foundations. *International Journal of Approximate Reasoning*, *28*, 91-124.

Fan, H.-Y., & Lampinen, J. (2003). A Trigonometric Mutation Operation to Differential Evolution. *Journal of Global Optimization*, *27*, 105-129.

Gerig, G., Welti, D., Guttman, C. R. G., Colchester, A. C. F., & Szekely, G. (2000). Exploring the discrimination power of the time domain for segmentation and characterisation of active lesions in serial MR data. *Medical Image Analysis*, *4*, 31-42.

Haenni, R., & Lehmann, N. (2002). Resource bounded and anytime approximation of belief function computations. *International Journal of Approximate Reasoning*, *31*, 103-154.

Kulasekere, E. C., Premaratne, K., Dewasurendra, D. A., Shyu, M.-L., Bauer, P. H. (2004). Conditioning and updating evidence. *International Journal of Approximate Reasoning*, *36*, 75-108.

Jones, A. L., Beynon, M. J., Holt, C. A., & Roy, S. (2006). A novel approach to the exposition of the temporal development of post-op osteoarthritic knee subjects. *Journal of Biomechanics*, *39*(13), 2512 – 2520.

Murphy, C. K. (2000). Combining belief functions when evidence conflicts. *Decision Support Systems*, *29*, 1-9.

Pearl, J. (1990). Reasoning with belief functions: An analysis of comparability. *International Journal of Approximate Reasoning*, *4*, 363-390.

Petit-Renaud, S., & Denœux, T. (2004). Nonparametric regression analysis of uncertain and imprecise data using belief functions. *International Journal of Approximate Reasoning*, *35*, 1-28.

Scotney, B., & McClean, S. (2003). Database aggression of imprecise and uncertain evidence. *Information Sciences*, *155*, 245-263.

Shafer, G. A. (1976). *Mathematical theory of Evidence*. Princeton: Princeton University Press.

Smets, P. (1990). The Combination of Evidence in the Transferable belief Model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *12*(5), 447-458.

Storn, R., & Price, K. (1997). Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimisation*, *11*(4), 41-359.

Wu, W.-Z., Leung, Y., & Zhang, W.-X. (2002). Connections between rough set theory and Dempster-Shafer theory of evidence. *International Journal of General Systems*, *31*(4), 405-430.

Yang, M.-S., Chen, T.-C., & Wu, K.-L. (2003). Generalized belief function, plausibility function, and Dempster's combinations rule to fuzzy sets. *International Journal of Intelligent Systems*, *18*, 925-937.

KEY TERMS

Belief: In Dempster-Shafer theory, a positive function that represents the confidence that a proposition lies in a focal element or any subset of it.

Evolutionary Algorithm: An algorithm that incorporates aspects of natural selection or survival of the fittest.

Focal Element: In Dempster-Shafer theory Subset of the frame of discernment with a positive mass value associated with it.

Frame of Discernment: In Dempster-Shafer theory, a finite non-empty set of hypotheses.

Dempster-Shafer Theory: General methodology, also known as the theory of belief functions, its rudiments are closely associated with uncertain reasoning.

Mass Value: In Dempster-Shafer theory, the level of exact belief in a focal element.

Plausibility: In Dempster-Shafer theory, a positive function that represents the extent to which we fail to disbelieve a proposition described by a focal element.

Using Prior Knowledge in Data Mining

Francesca A. Lisi
Università degli Studi di Bari, Italy

INTRODUCTION

One of the most important and challenging problems in current Data Mining research is the definition of the *prior knowledge* that can be originated from the process or the domain. This contextual information may help select the appropriate information, features or techniques, decrease the space of hypotheses, represent the output in a most comprehensible way and improve the process. Ontological foundation is a precondition for efficient automated usage of such information (Chandrasekaran *et al.*, 1999). An *ontology* is a formal explicit specification of a shared conceptualization for a domain of interest (Gruber, 1993). Among other things, this definition emphasizes the fact that an ontology has to be specified in a language that comes with a *formal semantics*. Due to this formalization ontologies provide the machine interpretable meaning of concepts and relations that is expected when using a semantic-based approach (Staab & Studer, 2004). In its most prevalent use in Artificial Intelligence (AI), an ontology refers to an engineering artifact (more precisely, produced according to the principles of *Ontological Engineering* (Gómez-Pérez *et al.*, 2004)), constituted by a specific vocabulary used to describe a certain reality, plus a set of explicit assumptions regarding the intended meaning of the vocabulary words. This set of assumptions has usually the form of a *First-Order Logic* (FOL) theory, where vocabulary words appear as unary or binary predicate names, respectively called concepts and relations. In the simplest case, an ontology describes a hierarchy of concepts related by subsumption relationships; in more sophisticated cases, suitable axioms are added in order to express other relationships between concepts and to constrain their intended interpretation.

Ontologies can play several roles in Data Mining (Nigro *et al.*, 2007). In this chapter we investigate the use of ontologies as prior knowledge in Data Mining. As an illustrative case throughout the chapter, we choose the task of Frequent Pattern Discovery, it being the most representative product of the cross-fertilization among Databases, Machine Learning and Statistics that has given rise to Data Mining. Indeed it is central to an entire class of descriptive tasks in Data Mining among which Association Rule Mining (Agrawal *et al.*, 1993; Agrawal & Srikant, 1994) is the most popular. A pattern is considered as an intensional description (expressed in a given language $\mathcal{L}$) of a subset of a data set **r**. The support of a pattern is the relative frequency of the pattern within **r** and is computed with the evaluation function *supp*. The task of Frequent Pattern Discovery aims at the extraction of all frequent patterns, i.e. all patterns whose support exceeds a user-defined threshold of minimum support. The blueprint of most algorithms for Frequent Pattern Discovery is the *levelwise search* (Mannila & Toivonen, 1997). It is based on the following assumption: If a generality order $\geq$ for the language $\mathcal{L}$ of patterns can be found such that $\geq$ is monotonic w.r.t. *supp*, then the resulting space $(\mathcal{L}, \geq)$ can be searched breadth-first by starting from the most general pattern in $\mathcal{L}$ and alternating candidate generation and candidate evaluation phases.

BACKGROUND

The use of prior knowledge is already certified in Data Mining. Proposals for taking concept hierarchies into account during the discovery process are relevant to our survey because they can be considered a less expressive predecessor of ontologies, e.g. concept hierarchies are exploited to mine multiple-level association rules (Han & Fu, 1995; Han & Fu, 1999) or generalized association rules (Srikant & Agrawal, 1995). Both extend the levelwise search method so that patterns can refer to multiple levels of description granularity. They differ in the strategy used in visiting the concept hierarchy: the former visits the hierarchy top-down, the latter bottom-up.

The use of prior knowledge is also the distinguishing feature of Inductive Logic Programming (ILP) which was born at the intersection of Machine Learning (more precisely, Inductive Learning) and Logic Programming (Nienhuys-Cheng & de Wolf, 1997). Due to the com-

mon roots between Logic Programming and relational databases (Ceri et al., 1990), ILP has been more recently proposed as a logic-based approach to *Relational Data Mining* (Džeroski, 1996; Džeroski & Lavrač, 2001; Džeroski, 2002). Relational Data Mining is intended to overcome some limits of traditional Data Mining, e.g., in Association Rule Mining, by representing patterns and rules either as Datalog conjunctive queries (Dehaspe & De Raedt, 1997; Dehaspe & Toivonen, 1999) or as tree data structures (Nijssen & Kok, 2001; Nijssen & Kok, 2003). Note that none of these proposals for Association Rule Mining can exploit the semantic information conveyed by concept hierarchies because both adopt a syntactic generality relation for patterns and rules. More generally, prior knowledge in ILP is often not organized around a well-formed conceptual model such as ontologies.

MAIN FOCUS

In this section we consider the task of mining multiple-level association rules extended to the more complex case of having an ontology as prior knowledge and tackled with an ILP approach (Lisi & Malerba, 2004). We focus on the phase of Frequent Pattern Discovery.

The data set **r** must encompass both a database and an ontology, loosely or tightly integrated, so that the semantics can flow from the ontology to the database. To represent one such data set, a logical language that treats relational and structural knowledge in a unified way is necessary. Among the logical languages proposed by Ontological Engineering, *Description Logics* (DLs) are the most widely used (Baader *et al.*, 2007). The relationship between DLs and databases is rather strong. Several investigations have been carried out on the usage of DLs to formalize semantic data models. In these proposals concept descriptions are used to present the schema of a database. Unfortunately, DLs offer a weaker than usual query language. This makes also pure DLs inadequate as a Knowledge Representation (KR) framework in Data Mining problems that exploit ontological prior knowledge. Hybrid languages that integrate DLs and Datalog appear more promising. In (Lisi & Malerba, 2004), the data set **r** is a knowledge base represented according to the KR framework of $\mathcal{AL}$-log (Donini *et al.*, 1998), thus composed of a relational database in Datalog (Ceri *et al.*, 1990) and an ontology in the DL $\mathcal{ALC}$ (Schmidt-Schauss & Smolka, 1991).

The language $\mathcal{L}$ of patterns must be able to capture the semantics expressed in the background ontology. In (Lisi & Malerba, 2004), $\mathcal{L}$ is a language of unary conjunctive queries in $\mathcal{AL}$-log where the distinguished variable is constrained by the reference concept and the other variables are constrained by task-relevant concepts. All these concepts are taken from the underlying ontology, thus they convey semantics. Furthermore, the language is multi-grained in the sense that patterns that can be generated describe data at multiple levels of granularity. These levels refer to levels in the background ontology.

The generality order $\geq$ for the language $\mathcal{L}$ of patterns must be based on a semantic generality relation, i.e. a relation that checks whether a pattern is more general than another with respect to the prior knowledge. Up to now, most algorithms have focussed on a syntactical approach. However, the use of background knowledge would greatly improve the quality of the results. First, patterns and rules which are not equivalent from a syntactical point of view, may be semantically equivalent. Taking into account the semantical relationships between patterns improves the comprehensibility while decreasing the size of the discovered set of patterns. Second, while the use of prior knowledge increases the expressivity and therefore comes with a cost, it also allows to better exploit the benefits of some optimizations. In (Lisi & Malerba, 2004), the space of patterns is structured according to the semantic generality relation of $\mathcal{B}$-subsumption (Lisi & Malerba, 2003a) and searched by means of a downward refinement operator (Lisi & Malerba, 2003b). It has been proved that $\mathcal{B}$-subsumption fulfills the monotonicity requirement of the levelwise search (Lisi & Malerba, 2004). Note that the support of patterns is computed with respect to the background ontology.

This ILP approach to Frequent Pattern Discovery within the KR framework of $\mathcal{AL}$-log has been very recently extended to Cluster Analysis (Lisi & Esposito, 2007).

FUTURE TRENDS

Using ontologies as prior knowledge in Data Mining will become central to any application area where ontologies are playing a key role and Data Mining can be of help to users, notably the *Semantic Web* (Berners-Lee *et al.*, 2001). In particular, the main focus of

the present chapter can contribute to the emerging field of *Semantic Web Mining* (Stumme *et al.*, 2006) with logic-based approaches that fit naturally the KR frameworks adopted in the Semantic Web. At present, the W3C standard mark-up language for ontologies in the Semantic Web, OWL (Ontology Web Language), is based on very expressive DLs (Horrocks *et al.*, 2003). Mark-up languages for Semantic Web rules aimed at extending or integrating ontologies with Logic Programming are still under discussion but are likely to be inspired by $\mathcal{AL}$-log.

Besides the Semantic Web application context, more methodological work on ontologies as prior knowledge in Data Mining is needed. In this sense the work of Lisi and Malerba (2004) can be considered as a source of inspiration. Also it can be considered as a starting point for a broader investigation of implementation solutions that can overcome the efficiency and scalability issues of the ILP approach and pave the way to more extensive experiments on real-world data. We call *Onto-Relational Data Mining* this new frontier of Data Mining that extends Relational Data Mining to account for prior knowledge in the form of ontologies.

CONCLUSION

Data Mining was born at the intersection of three research areas: Databases, Machine Learning, and Statistics. The different perspectives offered by these areas lay strong emphases on different aspects of Data Mining. The emphasis of the Database perspective is on *efficiency* because this perspective strongly concerns the whole discovery process and huge data volume. The emphasis of the Machine Learning perspective is on *effectiveness* because this perspective is heavily attracted by substantive heuristics working well in data analysis although they may not always be useful. As for the Statistics perspective, its emphasis is on *validity* because this perspective cares much for mathematical soundness behind discovery methods. In this chapter we have adopted a broader AI perspective in order to address the issue of *meaningfulness*. In particular we have shown one way of achieving this goal, i.e. to have ontologies as prior knowledge. As a final remark, we would like to stress the fact that the resulting Data Mining algorithms must be able to actually deal with ontologies, i.e. without disregarding their nature of FOL theories equipped with a formal semantics.

REFERENCES

Agrawal, R., Imielinski, T. & Swami, A. (1993). Mining association rules between sets of items in large databases. In P. Buneman & S. Jajodia (Eds.), *Proc. of the ACM SIGMOD Conference on Management of Data* (pp. 207-216).

Agrawal, R. & Srikant, R. (1994). Fast Algorithms for Mining Association Rules. In J. B. Bocca, M. Jarke & C. Zaniolo (Eds.), *Proc. of the 20th Int. Conference on Very Large Databases* (pp. 487-499). Santiago, Chile.

Baader, F., Calvanese, D., McGuinness, D., Nardi, D., & Patel-Schneider, P.F. (Ed.) (2007). *The Description Logic Handbook: Theory, Implementation and Applications (2nd Edition)*. Cambridge University Press.

Berners-Lee, T., Hendler, J. & Lassila, O. (2001). The Semantic Web. *Scientific American*, May.

Ceri, S., Gottlob, G., & Tanca, L. (1990). *Logic Programming and Databases*. Springer.

Chandrasekaran, B., Josephson, J.R., & Benjamins, V.R. (1999). What are ontologies, and why do we need them? *IEEE Intelligent Systems*, 14(1), 20–26.

Dehaspe, L., & De Raedt, L. (1997). Mining association rules in multiple relations. In N. Lavrač & S. Džeroski (Ed.), *Inductive Logic Programming*, volume 1297 of Lecture Notes in Artificial Intelligence, 125-132. Springer-Verlag.

Dehaspe, L., & Toivonen, H. (1999). Discovery of frequent Datalog patterns. *Data Mining and Knowledge Discovery*, 3, 7–36.

Donini, F.M., Lenzerini, M., Nardi, D., & Schaerf, A. (1998). $\mathcal{AL}$-log: Integrating Datalog and Description Logics. *Journal of Intelligent Information Systems*, 10(3), 227–252.

Džeroski, S. (1996). Inductive logic programming and knowledge discovery in databases. In U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, & R. Uthurusamy (Eds.), *Advances in knowledge discovery and data mining*. (pp. 117–152). AAAI Press/The MIT Press.

Džeroski, S. (2002). Inductive logic programming approaches. In W. Klösgen, & J. M. Zytkow (Eds.), *Handbook of Data Mining and Knowledge Discovery*, (pp. 348-353). Oxford University Press, Oxford.

Džeroski, S., & Lavrač, N. (Ed.) (2001). *Relational Data Mining*. Springer.

Gómez-Pérez, A., Fernández-López, M., & Corcho, O. (2004). *Ontological Engineering*. Springer.

Gruber, T. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5, 99–220.

Han, J., & Fu, Y. (1995). Discovery of multiple-level association rules from large databases. In U. Dayal, P. Gray, & S. Nishio (Eds.), *Proc. of the 21st Int. Conference on Very Large Data Bases* (pp. 420–431). Morgan Kaufmann.

Han, J., & Fu, Y. (1999). Mining multiple-level association rules in large databases. *IEEE Transactions on Knowledge and Data Engineering*, 11(5), 798-804.

Horrocks, I., Patel-Schneider, P.F., & van Harmelen, F. (2003). From $\mathcal{SHIQ}$ and RDF to OWL: The making of a web ontology language. *Journal of Web Semantics*, 1(1), 7–26.

Lisi, F.A., & Esposito, F. (2007). On the Missing Link between Frequent Pattern Discovery and Concept Formation. In: S. Muggleton, R. Otero & A. Tamaddoni-Nezhad (Eds.), *Inductive Logic Programming*, volume 4455 of Lecture Notes in Artificial Intelligence (pp. 305-319). Springer: Berlin.

Lisi, F.A., & Malerba, D. (2003a). Bridging the Gap between Horn Clausal Logic and Description Logics in Inductive Learning. In A. Cappelli & F. Turini (Eds.), *AI*IA 2003: Advances in Artificial Intelligence*, volume 2829 of Lecture Notes in Artificial Intelligence (pp. 49–60). Springer.

Lisi, F.A., & Malerba, D. (2003b). Ideal Refinement of Descriptions in $\mathcal{AL}$-log. In T. Horvath & A. Yamamoto (Eds.), *Inductive Logic Programming*, volume 2835 of Lecture Notes in Artificial Intelligence (pp. 215–232). Springer.

Lisi, F.A., & Malerba, D. (2004). Inducing Multi-Level Association Rules from Multiple Relations. *Machine Learning*, 55, 175–210.

Maedche, A., & Staab, S. (2004). Ontology Learning. In S. Staab & R. Studer (Ed.), *Handbook on Ontologies*. Springer.

Mannila, H., & Toivonen, H. (1997). Levelwise search and borders of theories in knowledge discovery. *Data Mining and Knowledge Discovery*, 1(3), 241–258.

Nienhuys-Cheng, S.-H., & de Wolf, R. (1997). *Foundations of Inductive Logic Programming*, volume 1228 of Lecture Notes in Artificial Intelligence. Springer.

Nigro, H.O., González Císaro, A., & Xodo, D. (Ed.) (2007). *Data Mining with Ontologies: Implementations, Findings and Frameworks*. IGI Global.

Nijssen, S., & Kok, J.N. (2001). Faster association rules for multiple relations. In B. Nebel (Ed.), *Proc. of the 17th Int. Joint Conf. on Artificial Intelligence* (pp. 891–896). Morgan Kaufmann.

Nijssen, S., & Kok, J.N. (2003). Efficient frequent query discovery in FARMER. In N. Lavrač, D. Gamberger, H. Blockeel & L. Todorovski (Ed.), *Knowledge Discovery in Databases: PKDD 2003*, volume 2431 of Lecture Notes in Artificial Intelligence (pp. 350-362). Springer-Verlag.

Schmidt-Schauss, M. & Smolka, G. (1991). Attributive concept descriptions with complements. *Artificial Intelligence*, 48(1), 1–26.

Srikant, R., & Agrawal, R. (1995). Mining generalized association rules. In U. Dayal, P. Gray, & S. Nishio (Eds.), *Proc. of the 21st Int. Conf. on Very Large Data Bases* (pp. 407–419). Morgan Kaufmann.

Staab, S. & Studer, R. (Ed.) (2004). *Handbook on Ontologies*. International Handbooks on Information Systems. Springer.

Stumme, G., Hotho, A., & Berendt, B. (2006). Semantic Web Mining: State of the Art and Future Directions. *Journal of Web Semantics*, 4(2), 124-143.

KEY TERMS

Description Logics (DL): Family of Knowledge Representation languages which can be used to represent the terminological knowledge of an application domain in a structured and formally well-understood way. The name DL refers, on the one hand, to concept descriptions used to describe a domain and, on the other hand, to the logic-based semantics which can

be given by a translation into First-Order Logic. DLs were designed as an extension to frames and semantic networks, which were not equipped with formal logic-based semantics.

First-Order Logic (FOL): Formal deductive system that extends propositional logic by allowing quantification over individuals of a given domain (universe) of discourse.

Formal Semantics (of a Language): Given by a mathematical model that describes the possible computations described by the language.

Inductive Logic Programming (ILP): Machine Learning approach born at the intersection between Inductive Learning and Logic Programming. Its distinguishing feature is the use of prior knowledge during the learning process. Originally concerned with the induction of rules from examples within the Knowledge Representation framework of Horn Clausal Logic and with the aim of prediction, it has recently broadened its scope of interest in order to investigate Inductive Learning in First-Order Logic fragments other than Horn Clausal Logic (e.g., Description Logics) and induction goals other than prediction (e.g., description).

Knowledge Representation (KR): AI research area that deals with the problem of representing, maintaining and manipulating knowledge about an application domain. Since virtually all AI systems have to address this problem, KR is a crucial AI subfield.

Logic Programming: Declarative programming paradigm based on a subset of First-Order Logic known under the name of Horn Clausal Logic.

Meaningfulness: Semantic validity with respect to prior knowledge.

Ontology: Formal explicit specification of a shared conceptualization for a domain of interest.

Ontological Engineering: The branch of Knowledge Engineering that studies ontologies as engineering artifact. It refers to the set of activities that concern the ontology development process, the ontology life cycle, the methods and methodologies for building ontologies, and the tool suites and languages that support them.

Relational Data Mining: The branch of Data Mining that studies methods and techniques having the relational data model as an invariant of the discovery process.

Semantic Web: An extension of the current World Wide Web in which information is given well-defined meaning, better enabling computers and people to work in cooperation.

Semantic Web Mining: Cross-fertilization between Web Mining and the Semantic Web in order to improve the results of Web Mining by exploiting the new semantic structures in the web, as well as to use Web Mining to help build the Semantic Web.

Utilizing Fuzzy Decision Trees in Decision Making

Malcolm J. Beynon,
Cardiff University, UK

INTRODUCTION

The seminal work of Zadeh (1965), namely fuzzy set theory (FST), has developed into a methodology fundamental to analysis that incorporates vagueness and ambiguity. With respect to the area of data mining, it endeavours to find potentially meaningful patterns from data (Hu & Tzeng, 2003). This includes the construction of if-then decision rule systems, which attempt a level of inherent interpretability to the antecedents and consequents identified for object classification (See Breiman, 2001).

Within a fuzzy environment this is extended to allow a linguistic facet to the possible interpretation, examples including mining time series data (Chiang, Chow, & Wang, 2000) and multi-objective optimisation (Ishibuchi & Yamamoto, 2004). One approach to if-then rule construction has been through the use of decision trees (Quinlan, 1986), where the path down a branch of a decision tree (through a series of nodes), is associated with a single if-then rule. A key characteristic of the traditional decision tree analysis is that the antecedents described in the nodes are crisp, where this restriction is mitigated when operating in a fuzzy environment (Crockett, Bandar, Mclean, & O'Shea, 2006).

This chapter investigates the use of fuzzy decision trees as an effective tool for data mining. Pertinent to data mining and decision making, Mitra, Konwar and Pal (2002) succinctly describe a most important feature of decision trees, crisp and fuzzy, which is their capability to break down a complex decision-making process into a collection of simpler decisions and thereby, providing an easily interpretable solution.

BACKGROUND

The development of fuzzy decision trees brings a linguistic form to the if-then rules constructed, offering a concise readability in their findings (see Olaru & Wehenkel, 2003). Examples of their successful application include in the areas of optimising economic dispatch (Roa-Serpulveda, Herrera, Pavez-Lazo, Knight, & Coonick, 2003) and the antecedents of company audit fees (Beynon, Peel, & Yang, 2004). Even in application based studies, the linguistic formulisation to decision making is continually investigated (Chakraborty, 2001; Herrera, Herrera-Viedma, & Martinez, 2000).

Appropriate for a wide range of problems, the fuzzy decision trees approach (with linguistic variables) allows a representation of information in a direct and adequate form. A linguistic variable is described in Herrera, Herrera-Viedma, & Martinez (2000), highlighting it differs from a numerical one, with it instead using words or sentences in a natural or artificial language. It further serves the purpose of providing a means of approximate characterization of phenomena, which are too complex, or too ill-defined to be amenable to their description in conventional quantitative terms.

The number of elements (words) in a linguistic term set which define a linguistic variable determines the granularity of the characterisation. The semantic of these elements is given by fuzzy numbers defined in the [0, 1] interval, which are described by their membership functions (MFs). Indeed, it is the role played by, and the structure of, the MFs that is fundamental to the utilization of FST related methodologies (Medaglia, Fang, Nuttle, & Wilson, 2002; Reventos, 1999). In this context, DeOliveria (1999) noted that fuzzy systems have the important advantage of providing an insight on the linguistic relationship between the variables of a system.

Within an inductive fuzzy decision tree, the underlying knowledge related to a decision outcome can be can be represented as a set of fuzzy if-then decision rules, each of the form:

If (A_1 is $T_{i_1}^1$) and (A_2 is $T_{i_2}^2$) … and (A_k is $T_{i_k}^k$) then C is C_j,

where $A_1, A_2, .., A_k$ and C are linguistic variables in the multiple antecedents (A_is) and consequent (C) statements, respectively, and $T(A_k) = \{T_1^k, T_2^k, .. T_{S_i}^k\}$ and $\{C_1, C_2, \ldots, C_L\}$ are their concomitant linguistic terms. Each linguistic term T_j^k is defined by the MF $\mu_{T_j^k}(x)$, which transforms a value in its associated domain to a grade of membership value to between 0 and 1. The MFs, $\mu_{T_j^k}(x)$ and $\mu_{C_j}(y)$, represent the grade of membership of an object's attribute value for A_j being T_j^k and C being C_j, respectively (Wang, Chen, Qiang, & Ye, 2000; Yuan & Shaw, 1995).

Different types of MFs have been proposed to describe fuzzy numbers, including triangular and trapezoidal functions (Lin & Chen, 2002; Medaglia, Fang, Nuttle, & Wilson, 2002). Yu and Li (2001) highlight that MFs may be (advantageously) constructed from mixed shapes, supporting the use of piecewise linear MFs. A general functional form of a piecewise linear MF (in the context of a linguistic term), is given by:

$$\mu_{T_j^k}(x)\begin{cases} 0 & \text{if } x \le \alpha_{j,1} \\ 0.5\dfrac{x-\alpha_{j,1}}{\alpha_{j,2}-\alpha_{j,1}} & \text{if } \alpha_{j,1} < x \le \alpha_{j,2} \\ 0.5+0.5\dfrac{x-\alpha_{j,2}}{\alpha_{j,3}-\alpha_{j,2}} & \text{if } \alpha_{j,2} < x \le \alpha_{j,3} \\ 1 & \text{if } x = \alpha_{j,3} \\ 1-0.5\dfrac{x-\alpha_{j,3}}{\alpha_{j,4}-\alpha_{j,3}} & \text{if } \alpha_{j,3} < x \le \alpha_{j,4} \\ 0.5-0.5\dfrac{x-\alpha_{j,4}}{\alpha_{j,5}-\alpha_{j,4}} & \text{if } \alpha_{j,4} < x \le \alpha_{j,5} \\ 0 & \text{if } \alpha_{j,5} < x \end{cases}$$

with the respective *defining values* in list form, $[\alpha_{j,1}, \alpha_{j,2}, \alpha_{j,3}, \alpha_{j,4}, \alpha_{j,5}]$. A Visual representation of this MF form is presented in Figure 1, which elucidates its general structure along with the role played by the respective sets of defining values.

The MF form presented in Figure 1 shows how the value of a MF is constrained within 0 and 1. The implication of the specific defining values is also illustrated, including the idea of associated support, the domain $[\alpha_{j,1}, \alpha_{j,5}]$ in Figure 1. Further, the notion of dominant support can also be considered, where a MF is most closely associated with an attribute value, the domain $[\alpha_{j,2}, \alpha_{j,4}]$ in Figure 1 (see Kovalerchuk & Vityaev, 2000).

Also included in Figure 1, using dotted lines are neighbouring MFs (linguistic terms), which collectively would define a linguistic variable, describing a continuous attribute. To circumvent the influence of expert opinion in analysis, the construction of the MFs should be automated. On this mater, DeOliveria (1999) considers the implication of Zadeh's principle of incompatibility - that is, as the number of MFs increase, so the precision of the system increases, but at the expense of decreasing relevance.

MAIN THRUST

Formulization of Fuzzy Decision Tree

The first fuzzy decision tree reference is attributed to Chang and Pavlidis (1997). A detailed description on the concurrent work of fuzzy decision trees is presented in Olaru & Wehenkel (2003). It highlights how methodologies include the fuzzification of a crisp decision tree post its construction (Pal & Chakraborty, 2001), or approaches that directly integrate fuzzy techniques during the tree-growing phase. The latter formulization is described here, with the inductive method proposed by Yuan and Shaw (1995) considered, based on measures of cognitive uncertainties.

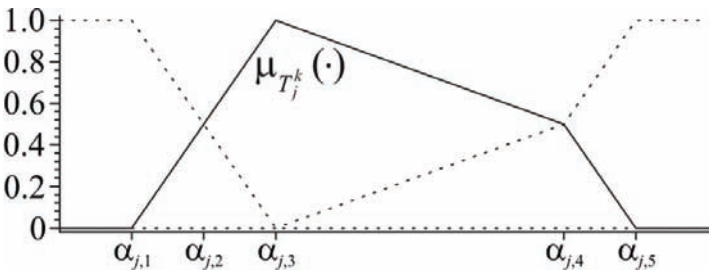
A MF $\mu(x)$ from the set describing a fuzzy linguistic variable Y defined on X, can be viewed as a possibility distribution of Y on X, that is $\pi(x) = \mu(x)$, for all $x \in X$ the values taken by the objects in U (also normalized so $\max_{x \in X}\pi(X) = 1$). The possibility measure $E_\alpha(Y)$ of ambiguity is defined by $E_\alpha(Y) = g(\pi) = \sum_{i=1}^{n}(\pi_i^* - \pi_{i+1}^*)\ln[i]$, where $\pi^* = \{\pi_1^*, \pi_2^*, \ldots, \pi_n^*\}$ is the permutation of the normalized possibility distribution $\pi = \{\pi(x_1), \pi(x_2), \ldots, \pi(x_n)\}$, sorted so that $\pi_i^* \ge \pi_{i+1}^*$ for $i = 1, .., n$, and $\pi_{i+1}^* = 0$. In the limit, if $\pi_2^* = 0$, then $E_\alpha(Y) = 0$, indicates no ambiguity, whereas if $\pi_n^* = 1$, then $E_\alpha(Y) = \ln[n]$, which indicates all values are fully possible for Y, representing the greatest ambiguity.

The ambiguity of attribute A (over the objects $u_1, .., u_m$) is given as:

$$E_\alpha(A) = \frac{1}{m}\sum_{i=1}^{m} E_\alpha(A(u_i)),$$

where

Figure 1. Definition of a piecewise linear MF (including defining values)



$$E_\alpha(A(u_i)) = g(\mu_{T_s}(u_i) \Big/ \max_{1 \le j \le s}(\mu_{T_j}(u_i))),$$

with $T_1, \ldots, T_s$ the linguistic terms of an attribute (antecedent) with m objects. When there is overlapping between linguistic terms (MFs) of an attribute or between consequents, then ambiguity exists.

For all $u \in U$, the intersection $A \cap B$ of two fuzzy sets is given by $\mu_{A \cap B} = \min[\mu_A(u), \mu_B(u)]$. The fuzzy subsethood $S(A, B)$ measures the degree to which A is a subset of B, and is given by,

$$S(A, B) = \sum_{u \in U} \min(\mu_A(u), \mu_B(u)) \Big/ \sum_{u \in U} \mu_A(u).$$

Given fuzzy evidence E, the possibility of classifying an object to the consequent C_i can be defined as,

$$\pi(C_i|E) = S(E, C_i) / \max_j S(E, C_j),$$

where the fuzzy subsethood $S(E, C_i)$ represents the degree of truth for the classification rule ('if E then C_i'). With a single piece of evidence (a fuzzy number for an attribute), then the classification ambiguity based on this fuzzy evidence is defined as: $G(E) = g(\pi(C|E))$, which is measured using the possibility distribution $\pi(C|E) = (\pi(C_1|E), \ldots, \pi(C_L|E))$.

The classification ambiguity with fuzzy partitioning $P = \{E_1, \ldots, E_k\}$ on the fuzzy evidence F, denoted as $G(P|F)$, is the weighted average of classification ambiguity with each subset of partition:

$$G(P|F) = \sum_{i=1}^{k} w(E_i|F) G(E_i \cap F),$$

where $G(E_i \cap F)$ is the classification ambiguity with fuzzy evidence $E_i \cap F$, and where $w(E_i|F)$ is the weight which represents the relative size of subset $E_i \cap F$ in F:

$$w(E_i|F) = \sum_{u \in U} \min(\mu_{E_i}(u), \mu_F(u)) \Big/ \sum_{j=1}^{k} \left(\sum_{u \in U} \min(\mu_{E_j}(u), \mu_F(u)) \right).$$

In summary, attributes are assigned to nodes based on the lowest level of classification ambiguity. A node becomes a leaf node if the level of subsethood is higher than some truth value β assigned to the whole of the fuzzy decision tree. The classification from the leaf node is to the decision group with the largest subsethood value. The truth level threshold β controls the growth of the tree; lower β may lead to a smaller tree (with lower classification accuracy), higher β may lead to a larger tree (with higher classification accuracy).

Illustrative Application of Fuzzy Decision Tree Analysis to Audit Fees Problem

The data mining of an audit fees model may be useful to companies in assessing whether the audit fee they are paying is reasonable. In this analysis, a sample of 120 UK companies is used for training (growing) a fuzzy decision tree (Beynon, Peel, & Tang, 2004).

The variables used in this study, are the decision attribute, AFEE: Audit fee (£000's) and condition attributes; SIZE: Sales (£000's), SUBS: Number of subsidiaries, FORS: ratio of foreign to total subsidiaries, GEAR: Ratio of debt to total assets, CFEE: Consultancy fees (£000's), TOTSH: proportion of shares held by directors and substantial shareholders, BIG6: 1 = Big 6 auditor, zero otherwise, LOND: 1 = audit office in London, zero otherwise.

The construction of the MFs (their defining values), which define the linguistic terms for a continuous attribute (linguistic variable) is based on an initial intervalisation of their domains. Then the subsequent

construction of estimated distributions of the values in each defined interval, see Figure 2.

In Figure 2(left), the decision attribute AFEE is intervalized, and three estimated distributions constructed (using probability distribution functions – *pdf*s). Using the centre of area of a distribution to denote an $a_{j,3}$ defining values and the interval boundary values the respective $a_{j,2}$ and $a_{j,4}$, the concomitant piecewise linear MFs are reported in Figure 2(right). Their labels, L, M and H, denote the linguistic terms, low, medium and high, for AFEE. Similar sets of MFs (linguistic variables) can be defined for the six continuous condition attributes, see Figure 3.

The series of linguistic variables (Figure 2 and 3) and the binary variables BIG and LOND are fundamental to the construction of a fuzzy decision tree for the audit fee model. The inductive fuzzy decision tree method is applied to the data on audit fees previously described, with a minimum "truth level" (β) of 0.80 required for a node to be a leaf (end) node, the complete tree is reported in Figure 4.

The fuzzy decision tree in Figure 4 indicates that a total of 21 leaf nodes (fuzzy if-then rules) were established. To illustrate, the rule labelled **R1** interprets to:

If SALES = L, CFEE = L and SUBS = M, then AFEE = M with truth level 0.9273

Or (equivalently):

For a company with low sales, low consultancy fees and a medium number of subsidiaries, then a low level of audit fees is expected with minimum truth level 0.9273

The fuzzy decision tree reported in Figure 4 as well as elucidating the relationship between audit fees (AFEE) and company characteristics can also be utilised to make predictions (matching) on the level of AFEE on future companies, not previously considered. The procedure (Wang et al., 2000), is as follows: *i*) Matching starts from the root node and ends at a leaf node along the branch of the maximum membership, *ii*) If the maximum membership at the node is not unique, matching proceeds along several branches, and *iii*) The decision class with the maximum degree of truth from the leaf nodes is then assigned the classification (e.g., L, M or H) for the associated rule.

FUTURE TRENDS

The ongoing characterization of data mining techniques in a fuzzy environment identifies the influence of its incorporation. The theoretical development of fuzzy set theory will, as well as explicit fuzzy techniques, continue to bring the findings from analysis nearer to the implicit human decision making.

Central to this is the linguistic interpretation associated with fuzzy related membership functions (MFs). There is no doubt their development will continue, including their structure and number of MFs to describe a linguistic variable. This is certainly true for the specific area of fuzzy decision trees, where the fuzzy if-then rules constructed offer a succinct elucidation to the data mining of large (and small) data sets.

The development of fuzzy decision trees will also include the alternative approaches to pruning, already considered in the non-fuzzy environment, whereby the branches of a decision tree may be reduced in size to offer even more general results. Recent attention has been towards the building of ensemble of classifiers. Here this relates to random forests (Breiman, 2001), whereby a number of different decision trees are constructed, subject to a random process. This can also include the use of techniques such as neural networks (Zhou & Jiang, 2003).

Figure 2. Estimated distributions and MFs (including defining values) for AFEE

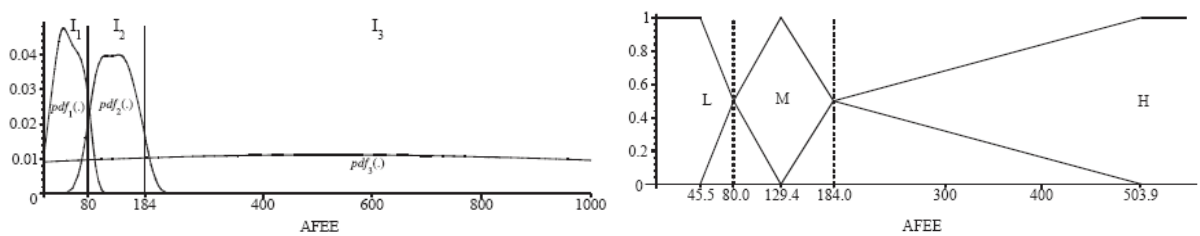


Figure 3. Sets of MFs for the six continuous condition attributes

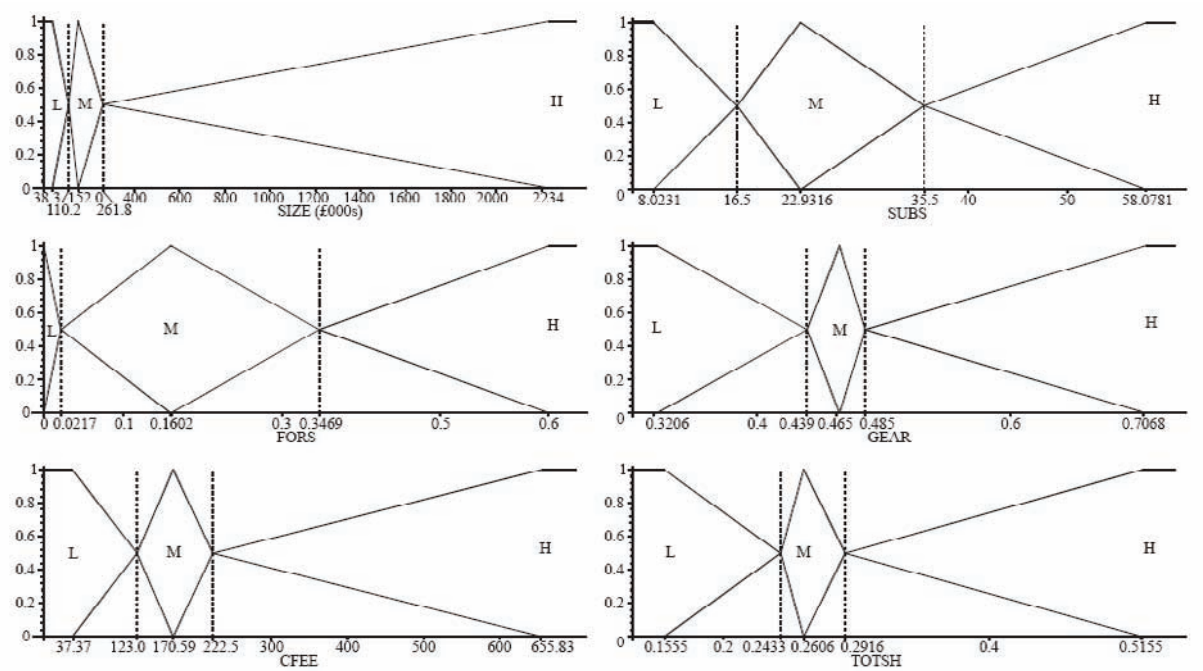
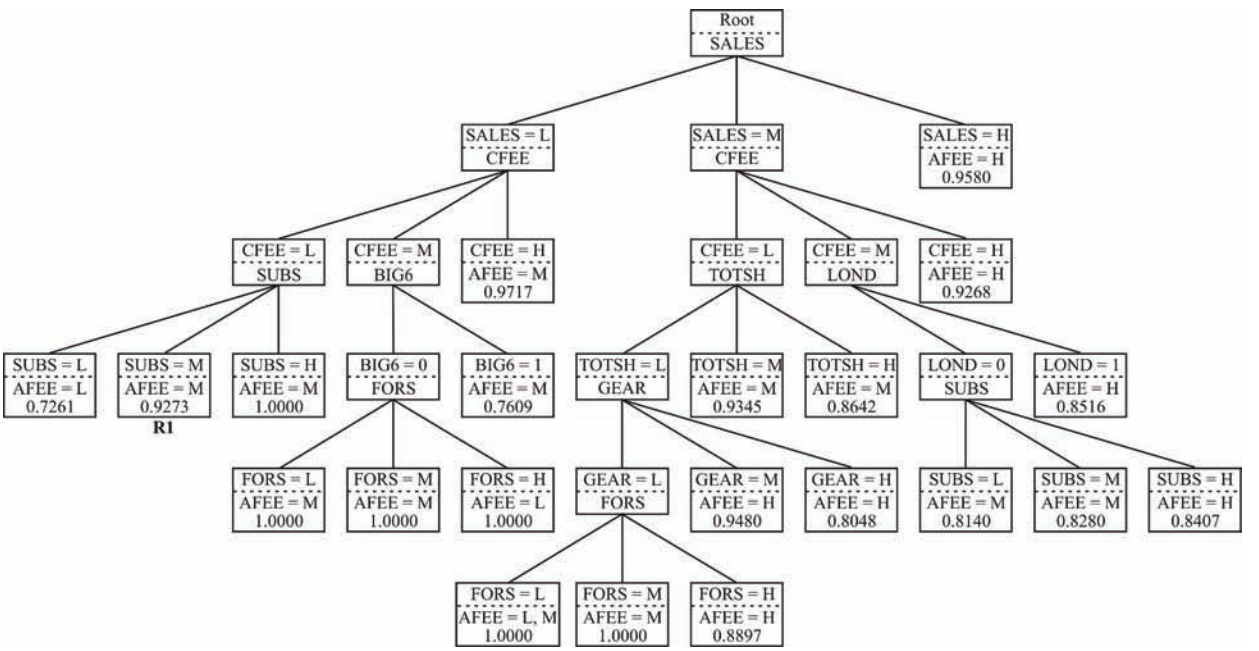


Figure 4. Complete fuzzy decision tree for audit fees problem



CONCLUSION

Within the area of data mining, the increased computational power (speed) available has reduced the perceived distance (relationship) between the original data and the communication of the antecedents and consequent to the relevant decision problem. As a tool for data mining, fuzzy decision trees exhibit the facets necessary for meaningful analysis. These include the construction of if-then decision rules to elucidate the warranted relationship. The fuzzy environment offers a linguistic interpretation to the derived relationship.

REFERENCES

Beynon, M. J., Peel, M. J., & Tang, Y.-C. (2004b). The Application of Fuzzy Decision Tree Analysis in an Exposition of the Antecedents of Audit Fees. *OMEGA - International Journal of Management Science, 32*(2), 231-244.

Breiman, L. (2001). Statistical modeling: the two cultures. *Statistical Science, 16*(3), 199-231.

Chakraborty, D. (2001). Structural quantization of vagueness in linguistic expert opinion in an evaluation programme. *Fuzzy Sets and Systems, 119*, 171-186

Chang, R. L. P., & Pavlidis, T. (1977). Fuzzy decision tree algorithms. *IEEE Transactions Systems Man and Cybernetics, SMC-7*(1), 28-35.

Chiang, D.-A., Chow, L. R., & Wang, Y.-F. (2000). Mining time series data by a fuzzy linguistic summary system. *Fuzzy Sets and Systems, 112*, 419-32.

Crockett, K., Bandar, Z., Mclean D., & O'Shea, J. (2006). On constructing a fuzzy inference framework using crisp decision trees. *Fuzzy Sets and Systems, 157*, 2809-2832.

DeOliveria, J. V. (1999). Semantic constraints for membership function optimization. *IEEE Transactions on Systems, Man and Cybernetics—Part A: Systems and Humans, 29*(1), 128-138.

Herrera, F., Herrera-Viedma, E., & Martinez, L. (2000). A fusion approach for managing multi-granularity linguistic term sets in decision making. *Fuzzy Sets and Systems, 114*(1), 43-58.

Hu, Y.-C., & Tzeng, G.-H. (2003). Elicitation of classification rules by fuzzy data mining. *Engineering Applications of Artificial Intelligence, 16*, 709-716.

Ishibuchi, H., & Yamamoto, T. (2004). Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining. *Fuzzy Sets and Systems, 141*, 59-88.

Kovalerchuk, B., & Vityaev, E. (2000). *Data mining in finance: advances in relational and hybrid methods*. Dordrecht, Kluwer Academic Publishers.

Lin, C.-C., & Chen, A.-P. (2002). Generalisation of Yang et. al.'s method of fuzzy programming with piecewise linear membership functions. *Fuzzy Sets and Systems, 132*, 346-352

Medaglia, A. L., & Fang, S.-C., Nuttle, H. L. W., & Wilson, J. R. (2002). An efficient and flexible mechanism for constructing membership functions. *European Journal of Operational Research, 139*, 84-95.

Mitra, S., Konwar, K. M., & Pal, S. K. (2002). Fuzzy Decision Tree, Linguistic Rules and Fuzzy Knowledge-Based Network: Generation and Evaluation. *IEEE Transactions on Systems, Man and Cybernetics - Part C: Applications and Reviews, 32*(4), 328-339.

Olaru, C., & Wehenkel, L. (2003). A complete fuzzy decision tree technique. *Fuzzy Sets and Systems, 138*, 221-254.

Pal, N. R., & Chakraborty, S. (2001). Fuzzy Rule Extraction From ID3-Type Decision Trees for Real Data. *IEEE Transactions on Systems, Man, and Cybernetics Part B, 31*(5), 745-754.

Quinlan, J. R. (1986). Introduction of Decision trees. *Machine Learning, 1*, 81-106.

Reventos, V. T. (1999). Interpreting membership functions: a constructive approach. *International Journal of Approximate Reasoning, 20*, 191-207.

Roa-Sepulveda, C. A., Herrera, M., Pavez-Lazo, B., Knight, U. G., & Coonick, A. H. (2000). A solution to the economic dispatch problem using decision trees. *Electric Power Systems Research, 56*, 255-259.

Wang, X., Chen, B., Qian, G., & Ye, F. (2000). On the optimization of fuzzy decision trees. *Fuzzy Sets and Systems, 112*, 117-125.

Yu, C.-S., & Li., H.-L. (2001). Method for solving quasi-concave and non-cave fuzzy multi-objective programming problems. *Fuzzy Sets and Systems*, *122*, 205-227.

Yuan, Y, & Shaw, M. J. (1995). Induction of fuzzy decision trees. *Fuzzy Sets and Systems*, *69*, 125-139.

Zadeh, L. A. (1965). Fuzzy Sets. *Information and Control*, *8*(3), 338-353.

Zhou, Z. H., & Jiang, Y. (2003). Medical diagnosis with C4.5 rule preceded by artificial neural networks ensemble. *IEEE Transactions on Information Technology in Biomedicine*, *7*, 37-42.

KEY TERMS

Antecedent: An antecedent is a driving factor in an event. For example, in the relationship "When it is hot, Mary buys an ice-scream", "it is hot" is the antecedent.

Branch: A single path down a decision tree, from root to a leaf node, denoting a single if-then rule.

Consequent: A consequent follows as a logical conclusion to an event. For example, in the relationship "When it is hot, Mary buys an ice-scream", "buys an ice-scream" is the consequent.

Decision Tree: A tree-like way of representing a collection of hierarchical rules that lead to a class or value.

Induction: A technique that infers generalizations from the information in the data.

Leaf: A node not further split – the terminal grouping – in a classification or decision tree.

Linguistic term: One of a set of linguistic terms which are subjective categories for a linguistic variable, each described by a membership function.

Linguistic variable: A variable made up of a number of words (linguistic terms) with associated degrees of membership towards them.

Node: A junction point in a decision tree, which describes a condition in an if-then rule.

Variable Length Markov Chains for Web Usage Mining

José Borges
School of Engineering, University of Porto, Portugal

Mark Levene
Birkbeck, University of London, UK

INTRODUCTION

Web usage mining is usually defined as the discipline that concentrates on developing techniques that model and study users' Web navigation behavior by means of analyzing data obtained from user interactions with Web resources; see (Mobasher, 2006; Liu, 2007) for recent reviews on web usage mining. When users access Web resources they leave a trace behind that is stored in log files, such traces are called *clickstream* records. Clickstream records can be preprocessed into time-ordered sessions of sequential clicks (Spiliopoulou et al., 2003), where a *user session* represents a *trail* the user followed through the Web space. The process of session reconstruction is called *sessionizing*.

Understanding user Web navigation behavior is a fundamental step in providing guidelines on how to improve users' Web experience. In this context, a model able to represent usage data can be used to induce frequent navigation patterns, to predict future user navigation intentions, and to provide a platform for adapting Web pages according to user specific information needs (Anand et al., 2005; Eirinaki et al., 2007). Techniques using association rules (Herlocker et al., 2004) or clustering methods (Mobasher et al., 2002) have been used in this context. Given a set of transactions clustering techniques can be used, for example, to find user segments, and association rule techniques can be used, for example, to find important relationships among pages based on the users navigational patterns. These methods have the limitation that the ordering of page views is not taken into consideration in the modeling of user sessions (Liu, 2007). Two methods that take into account the page view ordering are: tree based methods (Chen et al., 2003) used for prefetching Web resources, and Markov models (Borges et al., 2000; Deshpande et al., 2004) used for link prediction. Moreover, recent studies have been conducted on the use of visualization techniques for discovering navigational trends from usage data (Chen et al., 2007a; Chen et al., 2007b).

BACKGROUND

In (Mobasher, 2006) a review of Web usage mining methods was given and Markov models were discussed as one of the techniques used for the analysis of navigational patterns. In fact, *Markov models* provide an effective way of representing Web usage data, since they are based on a well established theory and provide a compact way of representing clickstream records. Markov models provide the means for predicting a user's next link choice based on his previous navigation trail (Dongshan et al., 2002; Deshpande et al., 2004), and as a platform for inducing user *frequent trails* (Borges et al., 2000).

In a first-order Markov model each Web page is represented by a state. In addition, a *transition probability* between two states represents the estimated probability (according to past usage) of viewing the two connected states in a sequence. Each user session is individually processed to count the number of times each page was visited and the number of times each pair of pages was viewed in a sequence, that is, the number of times a transition was followed. The model is built incrementally by processing the entire collection of user sessions. The transition probability between every pair of connected pages is estimated by the proportion of times the corresponding link was followed after viewing the anchor page. The ratio between the number of times a page was viewed and the total number of page views gives the probability estimate for a user choosing the corresponding page from the set of all pages in the site; we call this probability the *initial probability* of a state. The probability estimate of a trail is given by

the product of the initial probability of the first state in the trail and the probability of the traversed links, that is, the transition probabilities.

We note that, as an alternative for modeling a page as a state it is possible to group pages into categories, for example based on their content. In such a scenario each state corresponds to a Web page category, and state transitions model the user's navigation through page categories. In addition, in the case of Web sites composed of pages dynamically built from database queries, each state will correspond to a query. In this case the user's navigation through the content requests is being modeled rather than the navigation through the content that has been viewed. Finally, for Web sites in which page content changes frequently, mechanism that deal with concept drift, such as the use of a sliding window (Koychev, 2007), can be utilized in order to take into account the change of users' behavior over time.

MAIN FOCUS

Building a Variable Length Markov Chain

A first-order Markov model has the limitation of taking into account only the last viewed page when providing the next link choice prediction probability. Thus, it assumes that user navigation options are influenced only by the current page being viewed. To tackle this limitation a method based on building a sequence of higher-order Markov models and a technique to choose the best model to use in each case (Deshpande et al., 2004) has been proposed. However, we stress that the amount of navigation history a user takes into account when deciding which page to visit next, varies from site to site or even from page to page within a given site. Thus, a method that produces a single model representing the variable length history of pages is potentially valuable for studying user Web navigation behavior.

A Variable Length Markov Chain (VLMC) is a Markov model extension that allows variable length navigation history to be captured within a single model (Bejerano, 2004), and a method that transforms a first-order Markov model into a VLMC was presented in (Borges et al., 2005). The method makes use of a clustering-based *cloning* operation that duplicates states having in-paths inducing distinct conditional probabilities for subsequent link choices.

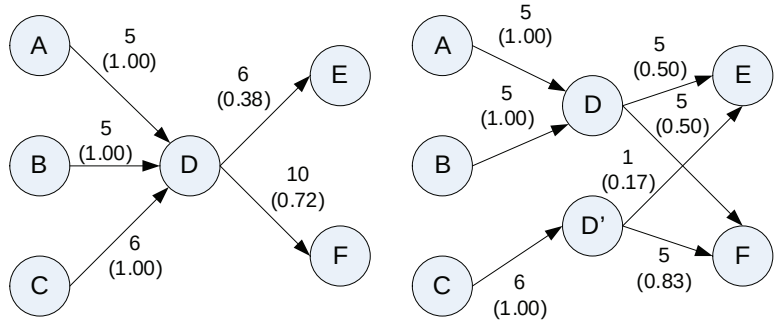
An n-order VLMC model is obtained by upgrading the corresponding previous-order model; for example, a second-order VLMC is obtained by upgrading a first-order model. The method evaluates one state at a time, measuring the accuracy with which n-order conditional probabilities are represented by the (n-1)-order model. From the input data, the n-order conditional probabilities are induced and compared to the (n-1)-order conditional probabilities represented by the model. If a state is shown to be inaccurate, its accuracy in representing the outlinks' transition probabilities can be increased by separating the in-paths to the state that correspond to different conditional probabilities. Thus, a state that is not accurately representing n-order conditional probabilities is cloned; the number of clones is determined by the required precision (set by a parameter) and a clustering technique that groups in-paths corresponding to similar conditional probabilities is used. As a result we obtain a model in which a transition probability between two states takes into account the path users followed to reach the anchor state prior to choosing the outlink corresponding to the transition.

We note that, when evaluating the accuracy of a state in a n-order model all n-length paths to that state are evaluated, thus, long sessions are pre-processed into (n+1)-grams whose corresponding conditional probabilities are evaluated in sequence. Sessions holding cycles are dealt with in the same way.

In Figure 1 we give a brief example to illustrate the method used to construct a first-order model from a collection of sessions and the resulting second-order model. On the left we present a collection of sessions and the corresponding frequency of occurrence, and, in the middle, we present the first-order model resulting from the sessions. Next to a link we show the number of times the link was followed and the corresponding estimated transition probability. According to the input data, the probability of the next link choice when viewing page D depends on the preceding page, thus, state D is set to be cloned. A clustering method identifies the estimated conditional probabilities from A and B to be closer, and therefore they are assigned to the same state; the conditional probabilities from C are kept separate. Transition probabilities from D and D' are more accurate in the resulting second-order model and in case higher accuracy is need more clones will be created.

Figure 1. An example of a collection of sessions (left), the corresponding first-order model (middle) and the second-order model (right)

Session	Frequency
A,D,E	3
A,D,F	2
B,D,E	2
B,D,F	3
C,D,E	1
C,D,F	5



EVALUATING VLMC MODELS

Several methods have been proposed and experiments conducted, to evaluate Variable Length Markov Chains in the context of Web usage mining. In (Borges et al., 2007a) two methods were presented that focus on testing the predictive power of a VLMC.

The first method is based on a $\chi 2$ (**Chi**-squared) statistical test that measures the distance between the probability distribution estimated by the model built from the full collection of sessions and the distribution estimated by the model induced from a subset of the collection. K-fold *cross validation* is used for building the two competing models: a model from (k-1) folds of the data set and a model from the full data set. The model obtained from the full data set represents the ground truth, the aim being to assess how well the model build from (k-1) folds can predict the unseen events in the remaining fold. A statistical test is used to compare the probability distributions of the predictions given by the two models concerned, and thus to evaluate a VLMC model's ability to generalize for unseen trails. Moreover, the result of the test can be used to evaluate the length of memory required to model users' navigation sessions.

The second method presented in (Borges et al., 2007a) evaluates a VLMC model's ability to predict the last page of a navigation session based on the preceding sequence of page views. The collection of input sessions is split into a training set and a test set, and a VLMC model is induced from the training set. Since results in previous experiments revealed a small variation among folds we decided to use a standard (70/30) training and test set split for this experiments. Then, for each session in the test set we use the VLMC model to estimate the probability of a user visiting the last page on the trail after having followed the preceding pages. The prediction accuracy is measured by the rank of the last page of the trail (the prediction target) among the ordered set of predictions given by the model; the higher the rank of the prediction target the better the prediction provided by the model. The average prediction ranking is returned as the overall metric.

In (Borges et al., 2007b) a technique is presented to measure the summarization ability of a VLMC model. The *Spearman footrule* metric (Fagin et al., 2003) is used to assess the accuracy with which a VLMC represents the information content of a collection of user sessions. For example, in order to measure how well a second-order VLMC model represents user trails having length four (i.e. a third-order model), the method collects from the input sessions all sequences of four pages ranked by frequency count. Then, an algorithm is used that gathers from the VLMC model the set of length-four trails ranked by the corresponding estimated traversal probability. The top-k elements of both rankings (where k is a parameter) are compared by means of the Spearman footrule metric, which is a metric used for comparing top-k ranked lists (Fagin et al., 2003).

We stress the importance of measuring both the summarization ability and the predictive power of a model. While a model that accurately summarizes the information content of a collection of user sessions can provide a potential platform for techniques focused on identifying users' frequent navigation patterns, a model showing strong predictive power provides the means to predict the next link choice of unseen navigation sessions and thus can be used for prefetching links or in adaptive Web site applications.

Results from extensive experiments conducted with real data sets were presented in (Borges et al., 2007a; Borges et al., 2007b) and show that, in general, a second or third-order model is sufficient to accurately represent a collection of user Web navigation sessions. The experimental results also provide evidence that there is a linear relationship between the predictive power and summarization ability of a VLMC model.

FUTURE TRENDS

Web usage mining research has focused on techniques that aim to deliver personalized, adaptive, effective and efficient Web services both from the point of view of the Web site owner and of the Web user. In the future, we expect methods to emerge that integrate Web usage mining techniques with Web content and Web structure mining techniques. For example, methods that are able to induce frequent patterns of users interested in a given topic or methods that are able to compare the induced navigation patterns with the underlying Web site topology in order to provide crucial insight to the Web site owner.

Another research direction that we expect to become very important in the near future is the problem of adapting the Web usage mining techniques to the context of Web 2.0, (Ankolekar et al., 2007). In fact, the way people use the Web is moving from that of taking a passive role, in which users access published content, to taking a more active role, in which users collaboratively produce and publish content. There are, for example, sites based on the Wiki concept, in which the content is dynamically kept up-to-date by a community of users, or sites in which users share and collaboratively tag multimedia material. These new paradigms of user interaction with Web resources pose fresh challenges to the Web usage mining community.

CONCLUSION

Web usage mining is an important topic within the more general field of web data mining. Several methods have been proposed to preprocess clickstream data into collections of user navigation sessions in a Web space as well as methods to model and analyze such data. Among the proposed methods we advocate the use of VLMC models because they provide a compact and powerful platform for Web usage analysis. An important property is the ability of a VLMC to take into account, for a given Web site, each page's amount of navigation history users consider when making the next link choice.

In this chapter we have reviewed recent research on methods that build VLMC models as well as methods devised to evaluate both the prediction power and the summarization ability of a VLMC model induced from a collection of navigation sessions. We believe that due to the well established concepts from Markov chain theory that underly VLMC models, they will be capable of providing support to cope with the new challenges in Web mining.

REFERENCES

Anand, S. and Mobasher, B. (2005). Intelligent Techniques for Web Personalization. *Lecture Notes in Computer Science, LNAI 3169*, 1-36.

Ankolekar, A., Krotzsch, M., Tran, T. and Vrandecic, D. (2007). The two cultures: mashing up web 2.0 and the semantic web. *Proceedings of the 16th international conference on World Wide Web*, 825-834.

Bejerano, G. (2004). Algorithms for variable length Markov chain modeling. *Bioinformatics*, 20(5), 788-789.

Borges, J. and Levene, M. (2000). Data mining of user navigation patterns. In *Web Usage Analysis and User Profiling*, LNAI 1836, 92-111.

Borges, J. and Levene, M. (2005). Generating dynamic higher-order Markov models in Web usage mining. LNAI 3721, 34-45.

Borges, J. and Levene, M. (2007a). Testing the predictive power of variable history Web usage. *Soft Computing*, 11(8), 717-727.

Borges, J. and Levene, M. (2007b). Evaluating variable length Markov chain models for analysis of user Web navigation sessions. *IEEE Transactions on Knowledge and Data Engineering*, 19 (4), 441-452.

Chen, X. and Zhang, X. (2003). A popularity-based prediction model for Web prefetching. *IEEE Computer*, 36(3), 63-70.

Chen, J., Zheng, T., Thorne, W., Huntley, D., Zayane, O.R., and Goebel, R. (2007a). Visualizing Web Navigation Data with Polygon Graphs. *Proceedings of the 11th International Conference Information Visualization*, 232-237.

Chen, J., Zheng, T., Thorne, W., Zayane, O.R., and Goebel, R. (2007b). Visual Data Mining of Web Navigational Data. *Proceedings of the 11th International Conference Information Visualization*, 649-656.

Deshpande, M. and G. Karypis, G. (2004). Selective Markov models for predicting Web page accesses. *ACM Transactions on Internet Technology*, 4(2), 163-184.

Dongshan, X. and Juni. S. (2002). A new Markov model for Web access prediction. *IEEE Computing in Science & Engineering*, 4(6), 34-39.

Eirinaki, M. and Vazirgiannis, M. (2007). Web site personalization based on link analysis and navigational patterns. *ACM Transactions on Internet Technology*, 7(4).

Fagin, R., Kumar, R., and Sivakumar, D. (2003). Comparing Top k Lists, *SIAM Journal on Discrete Mathematics*, 17(1), 134-160.

Herlocker, J.L., Konstan, J.A., Terveen, L.G. and Riedl, J. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22(1), 5-53

Koychev, I. (2007). Experiments with two approaches for tracking drifting concepts. *Serdica Journal of Computing*, 1(1), 27-44.

Liu, B. (2007). *Web Data Mining - Exploring Hyperlinks, Contents, and Usage Data*, Springer.

Mobasher, B, Dai, H., Luo,T., Nakagawa, M. (2002). Discovery and Evaluation of Aggregate Usage Profiles for Web Personalization. *Data Mining Knowledge Discovery*, 6(1), 61-82.

Mobasher, B. (2006). Web Usage Mining, In *Encyclopedia of Data Warehousing and Mining*, Idea Group Reference, 1216-1220.

Spiliopoulou, M., Mobasher, B., Berendt, B., and Nakagawa, M. (2003). A framework for the evaluation of session reconstruction heuristics in Web usage analysis. *IN-FORMS Journal on Computing*, 15(2), 171-190.

KEY TERMS

Clickstream: A recording of user clicks while browsing the Web.

Frequent Trail: A sequence of pages frequently followed by Web users.

Markov Model: A statistical model used to represent a stochastic process characterized by a set of states and a transition probability matrix.

Sessionizing: The process of reconstructing user sessions from clickstream data.

State Cloning: The process of duplicating selected states in order to accurately represent higher-order conditional probabilities.

Usage Mining: Automatic discovery of user navigation patterns from clickstream data.

User Session: A sequence of web page clicks followed by a user within a given time window.

Variable Length Markov Model: A Markov model extension that allows variable length navigation history to be captured.

Vertical Data Mining on Very Large Data Sets

William Perrizo
North Dakota State University, USA

Qiang Ding
Chinatelecom Americas, USA

Qin Ding
East Carolina University, USA

Taufik Abidin
North Dakota State University, USA

INTRODUCTION

Due to the rapid growth of the volume of data that are available, it is of importance and challenge to develop scalable methodologies and frameworks that can be used to perform efficient and effective data mining on large data sets. Vertical data mining strategy aims at addressing the scalability issues by organizing data in vertical layouts and conducting logical operations on vertical partitioned data instead of scanning the entire database horizontally in order to perform various data mining tasks.

BACKGROUND

The traditional horizontal database structure (files of horizontally structured records) and traditional scan-based data processing approaches (scanning files of horizontal records) are known to be inadequate for knowledge discovery in very large data repositories due to the problem of scalability. For this reason much effort has been put on sub-sampling and indexing as ways to address and solve the problem of scalability. However, sub-sampling requires that the sub-sampler know enough about the large dataset in the first place in order to sub-sample "representatively". That is, sub-sampling requires considerable knowledge about the data, which, for many large datasets, may be inadequate or non-existent. Index files are vertical structures. That is, they are vertical access paths to sets of horizontal records. Indexing files of horizontal data records does address the scalability problem in many cases, but it does so at the cost of creating and maintaining the index files separate from the data files themselves.

A new way to organize data is to organize them vertically, instead of horizontally. Data miners are typically interested in collective properties or predictions that can be expressed very briefly (e.g., a yes/no answer). Therefore, the result of a data mining query can be represented by a bitmap vector. This important property makes it possible to perform data mining directly on vertical data structures.

MAIN FOCUS

Vertical data structures, vertical mining approaches and multi-relational vertical mining will be explored in detail to show how vertical data mining works.

Vertical Data Structures

The concept of vertical partitioning has been studied within the context of both centralized and distributed database systems for a long time, yet much remains to be done (Winslett, 2002). There are great advantages of using vertical partitioning, for example, it makes hardware caching work really well; it makes compression easy to do; it may greatly increase the effectiveness of the I/O device since only participating fields are retrieved each time. The vertical decomposition of a relation also permits a number of transactions to be executed concurrently. Copeland & Khoshafian (1985) presented an attribute-level Decomposition Storage Model called DSM, similar to the Attribute Transposed File model (ATF) (Batory, 1979), that stores each column of a relational table into a separate table. DSM was shown to perform well. It utilizes surrogate keys to map individual attributes together, hence requiring a surrogate

key to be associated with each attribute of each record in the database. Attribute-level vertical decomposition is also used in Remotely Sensed Imagery (e.g., Landsat Thematic Mapper Imagery), where it is called Band Sequential (BSQ) format. Beyond attribute-level decomposition, Wong et al. (1985) presented the Bit Transposed File model (BTF), which took advantage of encoded attribute values using a small number of bits to reduce the storage space.

In addition to ATF, BTF, and DSM models, there has been other work on vertical data structuring, such as Bit-Sliced Indexes (BSI) (Chan & Ioannidis, 1998, O'Neil & Quass, 1997, Rinfret et al., 2001), Encoded Bitmap Indexes (EBI) (Wu & Buchmann, 1998; Wu, 1998), and Domain Vector Accelerator (DVA) (Perrizo et al., 1991).

A Bit-Sliced Index (BSI) is an ordered list of bitmaps used to represent values of a column or attribute. These bitmaps are called bit-slices, which provide binary representations of attribute's values for all the rows.

In the EBI approach, an encoding function on the attribute domain is applied and a binary-based bit-sliced index on the encoded domain is built. EBIs minimize the space requirement and show more potential optimization than binary bit-slices.

Both BSIs and EBIs are auxiliary index structures that need to be stored twice for particular data columns. As we know, even the simplest index structure used today incurs substantial increase in total storage requirements. The increased database size, in turn, translates into higher media and maintenance costs, and results in lower performance.

Domain Vector Accelerator (DVA) is a method to perform relational operations based on vertical bit-vectors. The DVA method performs particularly well for joins involving a primary key attribute and an associated foreign key attribute.

Vertical mining requires data to be organized vertically and be processed horizontally through fast, multi-operand logical operations, such as AND, OR, XOR, and complement. Predicate tree (P-tree[1]) (Ding et al., 2002) is one form of lossless vertical structure that meets this requirement. P-tree is suitable to represent numerical and categorical data and has been successfully used in various data mining applications, including classification (Khan et al., 2002; Ding et al., 2002; Perrizo et al., 2007), clustering (Denton et al., 2002), association rule mining (Ding et al., 2002; Imad et al., 2004), and outlier detection (Ren et al., 2004).

To convert a relational table of horizontal records to a set of vertical P-trees, the table has to be projected into columns, one for each attribute, retaining the original record order in each. Then each attribute column is further decomposed into separate bit vectors (called bit Sequential format, or bSQ format), one for each bit position of the values in that attribute. Each bit vector, i.e., bSQ file, is then compressed into a tree structure by recording the truth of the predicate "purely 1-bits" recursively on halves until purity is reached.

Figure 1 is an example of an 8×8 bSQ file and its corresponding P-tree (Ding et al., 2002). In this example, 36 is the number of 1's in the entire image, called root count. This root level is labeled level 0. The numbers 16, 7, 13, and 0 at the next level (level 1) are the 1-bit counts for the four major quadrants in raster order. Since the first and last level-1 quadrants are composed entirely of 1-bits (called pure-1 quadrants) and 0-bits (called pure-0 quadrants) respectively, sub-trees are not needed and these branches terminate. This pattern is continued recursively using the Peano or Z-ordering of the four sub-quadrants at each new level. Eventually, every branch terminates.

A variation of the P-tree data structure, the PM-tree, is a similar structure in which masks rather than counts

Figure 1. P-tree for a 8×8 bSQ file

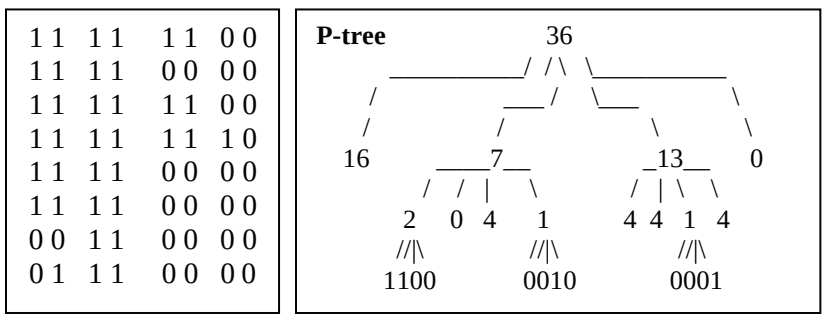
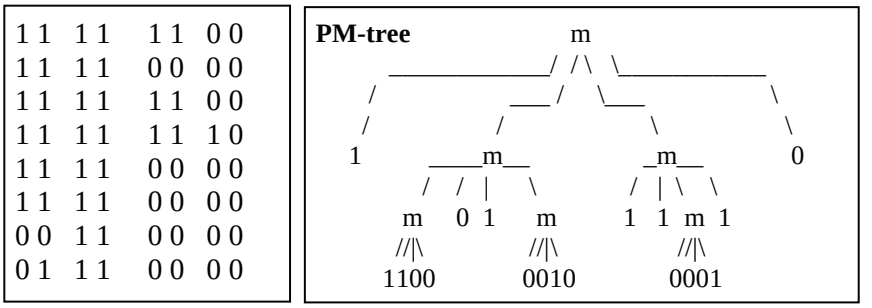


Figure 2. PM-tree



are used. In a PM-tree, a 3-value logic is used to represent pure-1, pure-0 and mixed quadrants (1 denotes pure-1, 0 denotes pure-0 and m denotes mixed). The PM-tree for the previous example is given in Figure 2.

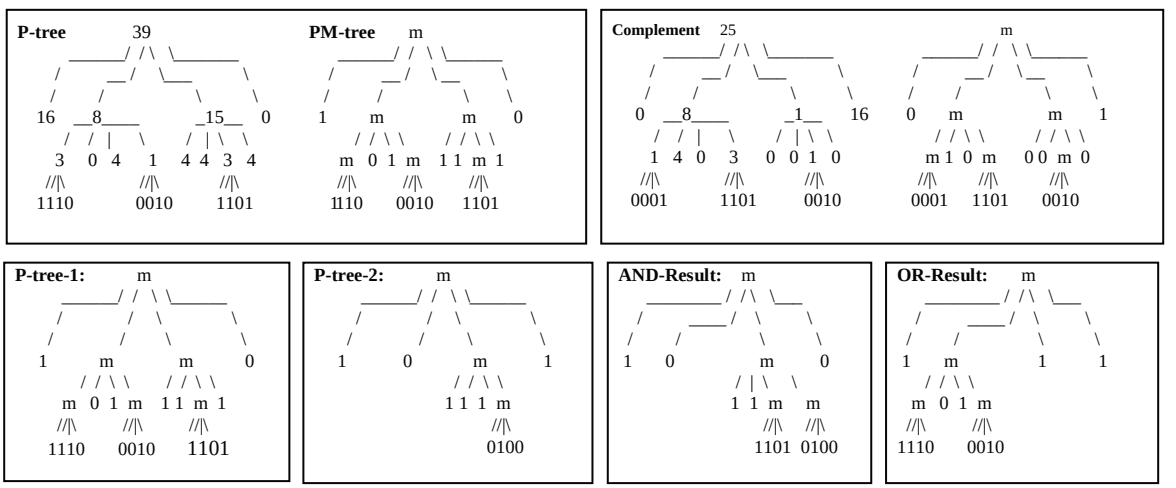
P-tree algebra includes three basic operations: complement, AND and OR. The complement of a P-tree can be constructed directly from the P-tree by simply complementing the counts at each level (subtracting from the pure-1 count at that level), as shown in the example below (Figure 3). The complement of a P-tree provides the 0-bit counts for each quadrant. P-tree AND/OR operations are also illustrated in Figure 3.

P-trees can be 1-dimensional, 2-dimensional, and multi-dimensional. If the data has a natural dimension (e.g., spatial data), the P-tree dimension is matched to the data dimension. Otherwise, the dimension can be chosen to optimize the compression ratio.

Vertical Mining Approaches

A number of vertical data mining algorithms have been proposed, especially in the area of association rule mining. Mining algorithms using the vertical format have been shown to outperform horizontal approaches in many cases. One example is the Frequent Pattern Growth algorithm using Frequent Pattern Trees introduced by Han in (Han et al., 2000). The advantages come from the fact that frequent patterns can be counted via the intersections of transaction_id sets, instead of using complex internal data structures. The horizontal approach, on the other hand, requires complex hash/search trees. Zaki & Hsiao (2002; 2003) introduced a vertical presentation called diffset, which keeps track of differences in the tidset of a candidate pattern from its generated frequent patterns. Diffset drastically reduces the memory required to store intermediate results; therefore, even in dense domains, the entire working

Figure 3. P-tree Algebra (Complement, AND and OR)



set of patterns of several vertical mining algorithms can be fit entirely in main-memory, facilitating the mining for very large database. Shenoy et al. (2000) proposes a vertical approach, called VIPER, for association rule mining of large databases. VIPER stores data in compressed bit-vectors and integrates a number of optimizations for efficient generation, intersection, counting, and storage of bit-vectors, which provides significant performance gains for large databases with a close to linear scale-up with database size.

P-trees have been applied to a wide variety of data mining areas. The efficient P-tree storage structure and the P-tree algebra provide a fast way to calculate various measurements for data mining task, such as support and confidence in association rule mining, information gain in decision tree classification, Bayesian probability values in Bayesian classification, etc.

P-trees have also been successfully used in many kinds of distance-based classification and clustering techniques. A computationally efficient distance metric called Higher Order Basic Bit Distance (HOBBit) (Khan et al., 2002) has been proposed based on P-trees. For one dimension, the HOBBit distance is defined as the number of digits by which the binary representation of an integer has to be right-shifted to make two numbers equal. For more than one dimension, the HOBBit distance is defined as the maximum of the HOBBit distances in the individual dimensions.

Since computers use binary systems to represent numbers in memory, bit-wise logical operations are much faster than ordinary arithmetic operations such as addition and multiplication of decimal numbers. Therefore, knowledge discovery algorithm utilizing vertical bit representation can accomplish its goals in a quick manner.

Multi-Relational Vertical Mining

Multi-Relational Data Mining (MRDM) is the process of knowledge discovery from relational databases consisting of multiple tables. The rise of several application areas in Knowledge Discovery in Databases (KDD) that is intrinsically relational has provided and continues to provide a strong motivation for the development of MRDM approaches. Since scalability has always been an important concern in the field of data mining, it is even more important in the multi-relational context, which is inherently more complex. From the perspective of database, multi-relational data mining usually involves one or more joins between tables, as is not the case for classical data mining methods. Until now, there is still lack of accurate, efficient, and scalable multi-relational data mining methods to handle large databases with complex schemas.

Databases are usually normalized for implementation reasons. However, for data mining workload, denormalizing the relations into a view may better represent the real world. In addition, if a view can be materialized by storing onto disks, it will be accessed much faster without being computed on the fly. Two alternative materialized view approaches, relational materialized view model and multidimensional materialized view model, can be utilized to model relational data (Ding, 2004).

In the relational model, a relational or extended-relational DBMS is used to store and manage data. A set of vertical materialized views can be created to encompass all the information necessary for data mining. Vertical materialized views can be generated directly from vertical representation of the original data. The transformation can be done in parallel by Boolean operations.

In the multidimensional materialized view model, multidimensional data are mapped directly to a data cube structure. The advantage of using a data cube is that it allows fast indexing by using offset calculation to precomputed data. The vertical materialized views of the data cube will require larger storage than that of the relational model. However, with the employment of the P-tree technology, there would not be much difference due to the compression inside the P-tree structure.

All the vertical materialized views can be easily stored in P-tree format. When encountering any data mining task, by relevance analysis and feature selection, all the relevant materialized view P-trees can be grabbed for data mining.

FUTURE TRENDS

Vertical data structure and vertical data mining will become more and more important. New vertical data structures will be needed for various types of data. There is great potential to combine vertical data mining with parallel mining as well as hardware. The scalability will become a very important issue in the area of data mining. The challenge of scalability is not just dealing with a large number of tuples but also handling with

high dimensionality (Fayyad, 1999, Han & Kamber, 2001).

CONCLUSION

Horizontal data structure has been proven to be inefficient for data mining on very large sets due to the large cost of scanning. It is of importance to develop vertical data structures and algorithms to solve the scalability issue. Various structures have been proposed, among which P-tree is a very promising vertical structure. P-trees have show great performance to process data containing large number of tuples due to the fast logical AND operation without scanning (Ding et al., 2002; Serazi et al., 2004)). Vertical structures, such as P-trees, also provide an efficient way for multi-relational data mining. In general, horizontal data structure is preferable for transactional data with intended output as a relation, and vertical data mining is more appropriate for knowledge discovery on very large data sets.

REFERENCES

Batory, D. S. (1979). On Searching Transposed Files. *ACM Transactions on Database Systems*, 4(4), 531-544.

Chan, C. Y. & Ioannidis, Y. (1998). Bitmap Index Design and Evaluation. *Proceedings of ACM SIGMOD International Conference on Management of Data*, 355-366.

Copeland, G. & Khoshafian, S. (1985). Decomposition Storage Model. *Proceedings of ACM SIGMOD International Conference on Management of Data*, 268-279.

Denton, A., Ding, Q., Perrizo, W., & Ding, Q. (2002). Efficient Hierarchical Clustering of Large Data Sets Using P-Trees. *Proceeding of International Conference on Computer Applications in Industry and Engineering*, 138-141.

Ding, Q., Ding, Q., & Perrizo, W. (2002). Decision Tree Classification of Spatial Data Streams Using Peano Count Trees. *Proceedings of ACM Symposium on Applied Computing*, 413-417.

Ding, Q. (2004). Multi-Relational Data Mining Using Vertical Database Technology. *Ph.D. Thesis, North Dakota State University.*

Ding, Q., Ding, Q., & Perrizo, W. (2002). Association Rule Mining on Remotely Sensed Images Using Ptrees. *Proceeding of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 66-79.

Ding, Q., Khan, M., Roy, A., & Perrizo, W. (2002). The P-tree Algebra. *Proceedings of ACM Symposium on Applied Computing*, 426-431.

Fayyad, U. (1999). Editorial, *SIGKDD Explorations*, 1(1), 1-3.

Han, J. & Kamber, M. (2006). *Data Mining: Concepts and Techniques*. San Francisco, CA: Morgan Kaufmann.

Han, J., Pei, J., & Yin, Y. (2000). Mining Frequent Patterns without Candidate Generation. *Proceedings of ACM SIGMOD International Conference on Management of Data*, 1-12.

Imad, R., Ren, D., Wu, W., Denton, A., Besemann, C., & Perrizo, W. (2006). Exploiting Edge Semantics in Citation Graph Data using Efficient, Vertical ARM. *Knowledge and Information Systems Journal*, 10(1), 57-91.

Khan, M., Ding, Q., & Perrizo, W. (2002). K-nearest Neighbor Classification on Spatial Data Stream Using Ptrees, *Proceeding of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 517-528.

O'Neil, P. & Quass, D. (1997). Improved Query Performance with Variant Indexes. *Proceedings of ACM SIGMOD International Conference on Management of Data*, 38-49.

Perrizo, W., Ding, Q., Khan, M., Denton, A. and Ding, Q. (2007). An Efficient Weighted Nearest Neighbour Classifier Using Vertical Data Representation. *International Journal of Business Intelligence and Data Mining*, 2(1), 64–78.

Perrizo, W., Gustafson, J., Thureen, D., & Wenberg, D. (1991). Domain Vector Accelerator for Relational Operations. *Proceedings of IEEE International Conference on Data Engineering*, 491-498.

Ren, D., Wang, B., & Perrizo, W. (2004). RDF: A Density-Based Outlier Detection Method using Vertical

Data Representation. *Proceedings of IEEE International Conference on Data Mining*, 503-506

Ren, D., Rahal, I., Perrizo, W., & Scott, K. (2004). A Vertical Outlier Detection Method with Local Pruning. *Proceedings of the ACM Conference on Information and Knowledge Management*, 279-284.

Ren, D., Rahal, I., & Perrizo, W. (2004). A Vertical Outlier Detection Method with Clusters as By-product. *Proceedings of the IEEE ICTAI, International Conference on Tools with Artificial Intelligence*, 22-29.

Rinfret, D., O'Neil, P., & O'Neil, E. (2001). Bit-Sliced Index Arithmetic. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 47-57.

Serazi, M., Perera, A., Ding, Q., Malakhov, V., Rahal, I., Pan, F., Ren, D., Wu, W., & Perrizo, W. (2004), DataMIME™. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 923-924.

Shenoy et al. (2000). Turbo-Charging Vertical Mining of Large Databases. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 22-33.

Winslett, M. (2002). David DeWitt Speaks Out. *ACM SIGMOD Record*, 31(2), 50-62.

Wong, H. K. T., Liu, H.-F., Olken, F., Rotem, D., & Wong. L. (1985). Bit Transposed Files. *Proceedings of International Conferences on Very Large Data Bases*, 448-457.

Wu, M-C. (1998). Query Optimization for Selections using Bitmaps. Technical Report, DVS98-2, DVS1, Computer Science Department, Technische Universitat Darmstadt.

Wu, M-C & Buchmann, A. (1998). Encoded Bitmap Indexing for Data Warehouses. *Proceedings of IEEE International Conference on Data Engineering*, 220-230.

Zaki, M. J., & Gouda, K. (2003). Fast Vertical Mining Using Diffsets. *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 326-335.

Zaki, M. J., & Hsiao, C-J. (2002). CHARM: An Efficient Algorithm for Closed Itemset Mining. *Proceedings of the SIAM International Conference on Data Mining*.

KEY TERMS

Association Rule Mining: The process of finding interesting association or correlation relationships among a large set of data items.

Data Mining: The application of analytical methods and tools to data for the purpose of identifying patterns and relationships such as classification, prediction, estimation, or affinity grouping.

HOBBit Distance: A computationally efficient distance metric. In one dimension, it is the number of digits by which the binary representation of an integer has to be right-shifted to make two numbers equal. In other dimension, it is the maximum of the HOBBit distances in the individual dimensions.

Multi-Relational Data Mining: The process of knowledge discovery from relational databases consisting of multiple tables.

Multi-Relational Vertical Mining: The process of knowledge discovery from relational databases consisting of multiple tables using vertical data mining approaches.

Predicate Tree (P-Tree): A lossless tree that is vertically structured and horizontally processed through fast multi-operand logical operations.

Vertical Data Mining (Vertical Mining): The process of finding patterns and knowledge from data organized in vertical formats, which aims to address the scalability issues.

ENDNOTE

[1] P-tree technology is patented by North Dakota State University (William Perrizo, primary inventor of record); patent number 6,941,303 issued September 6, 2005.

Video Data Mining

JungHwan Oh
University of Texas at Arlington, USA

JeongKyu Lee
University of Texas at Arlington, USA

Sae Hwang
University of Texas at Arlington, USA

INTRODUCTION

Data mining, which is defined as the process of extracting previously unknown knowledge and detecting interesting patterns from a massive set of data, has been an active research area. As a result, several commercial products and research prototypes are available nowadays. However, most of these studies have focused on corporate data — typically in an alpha-numeric database, and relatively less work has been pursued for the mining of multimedia data (Zaïane, Han, & Zhu, 2000). Digital multimedia differs from previous forms of combined media in that the bits representing texts, images, audios, and videos can be treated as data by computer programs (Simoff, Djeraba, & Zaïane, 2002). One facet of these diverse data in terms of underlying models and formats is that they are synchronized and integrated hence, can be treated as integrated data records. The collection of such integral data records constitutes a multimedia data set. The challenge of extracting meaningful patterns from such data sets has lead to research and development in the area of multimedia data mining. This is a challenging field due to the non-structured nature of multimedia data. Such ubiquitous data is required in many applications such as financial, medical, advertising and Command, Control, Communications and Intelligence (C3I) (Thuraisingham, Clifton, Maurer, & Ceruti, 2001). Multimedia databases are widespread and multimedia data sets are extremely large. There are tools for managing and searching within such collections, but the need for tools to extract hidden and useful knowledge embedded within multimedia data is becoming critical for many decision-making applications.

BACKGROUND

Multimedia data mining has been performed for different types of multimedia data: image, audio and video. Let us first consider image processing before discussing image and video data mining since they are related. Image processing has been around for some time. Images include maps, geological structures, biological structures, and many other entities. We have image processing applications in various domains including medical imaging for cancer detection, and processing satellite images for space and intelligence applications. Image processing has dealt with areas such as detecting abnormal patterns that deviate from the norm, and retrieving images by content (Thuraisingham, Clifton, Maurer, & Ceruti, 2001). The questions here are: *what* is image data mining and *how* does it differ from image processing? We can say that while image processing focuses on manipulating and analyzing images, image data mining is about finding useful patterns. Therefore, image data mining deals with making associations between different images from large image databases. One area of researches for image data mining is to detect unusual features. Its approach is to develop templates that generate several rules about the images, and apply the data mining tools to see if unusual patterns can be obtained. Note that detecting unusual patterns is not the only outcome of image mining; that is just the beginning. Since image data mining is an immature technology, researchers are continuing to develop techniques to classify, cluster, and associate images (Goh, Chang, & Cheng, 2001; Li, Li, Zhu, & Ogihara, 2002; Hsu, Dai, & Lee, 2003; Yanai, 2003; Müller & Pun, 2004). Image data mining is an area with applications in numerous domains including space, medicine, intelligence, and geoscience.

Mining video data is even more complicated than mining still image data. One can regard a video as a collection of related still images, but a video is a lot more than just an image collection. Video data management has been the subject of many studies. The important areas include developing query and retrieval techniques for video databases (Aref, Hammad, Catlin, Ilyas, Ghanem, Elmagarmid, & Marzouk, 2003). The question we ask yet again is what is the difference between video information retrieval and video mining? There is no clear-cut answer for this question yet. To be consistent with our terminology, we can say that finding correlations and patterns previously unknown from large video databases is video data mining.

MAIN THRUST

Even though we define video data mining as finding correlations and patterns previously unknown, the current status of video data mining remains mainly at the pre-processing stage, in which the preliminary issues such as video clustering, and video classification are being examined and studied for the actual mining. Only a very limited number of papers about finding any patterns from videos can be found. We discuss video clustering, video classification and pattern finding as follows.

Video Clustering

Clustering is a useful technique for the discovery of some knowledge from a dataset. It maps a data item into one of several clusters which are natural groupings for data items based on similarity metrics or probability density models (Mitra & Acharya, 2003). Clustering pertains to unsupervised learning, when data with class labels are not available. Clustering consists of partitioning data into homogeneous granules or groups, based on some objective function that maximizes the inter-cluster distances, while simultaneously minimizing the intra-cluster distances. Video clustering has some differences with conventional clustering algorithms. As mentioned earlier, due to the unstructured nature of video data, preprocessing of video data by using image processing or computer vision techniques is required to get structured format features. Another difference in video clustering is that the time factor should be considered while the video data is processed. Since video is a synchronized data of audio and visual data in terms of time, it is very important to consider the time factor. Traditional clustering algorithms can be categorized into two main types: partitional and hierarchical clustering (2003). Partitional clustering algorithms (i.e., *K-means* and *EM*) divide the patterns into a set of spherical clusters, while minimizing the objective function. Here the number of clusters is predefined. Hierarchical algorithms, on the other hand, can again be grouped as agglomerative and divisive. Here no assumption is made about the shape or number of clusters, and validity index is used to determine termination.

Two of the most popular partitional clustering algorithms are *K-means* and *Expectation Miximization (EM)*. In K-means, the initial centroids are selected, and each data item is classified to a cluster with the smallest distance. Based on the previous results, the cluster centroids are updated, and all corresponding data items are re-clustered until there is no centroid change. It is easily implemented, and provides a firm foundation of variances through the clusters. We can find the papers using the K-means algorithm for video clustering in the literature (Ngo, Pong, & Zhang, 2001). EM is a popular iterative refinement algorithm that belongs to the model-based clustering. It differs from the conventional K-means clustering algorithm in that each data point belongs to a cluster according to some weight or probability of membership. In other words, there are no strict boundaries between clusters. New means are computed based on weighted measures. It provides a statistical model for the data and is capable of handling the associated uncertainties. We can find the papers using the EM algorithm for video clustering in the literature (Lu, & Tan, 2002; Frey, & Jojic, 2003).

Hierarchical clustering methods create hierarchical nested partitions of the dataset, using a tree-structured dendogram and some termination criterion. Every cluster node contains child clusters; sibling clusters partition the points covered by their common parent. Such an approach allows exploring data on different levels of granularity. Hierarchical clustering methods are categorized into agglomerative (bottom-up) and divisive (top-down). An agglomerative clustering starts with one-point (singleton) clusters and recursively merges two or more of the most appropriate clusters. Divisive clustering starts with one cluster of all data points and recursively splits the most appropriate cluster. The process continues until a stopping criterion is achieved. The advantages of hierarchical clustering

include: embedded flexibility regarding the level of granularity, ease of handling of any forms of similarity or distance, and applicability to any attribute types. The disadvantages of hierarchical clustering are vagueness of termination criteria, and the fact that most hierarchical algorithms do not revisit constructed (intermediate) clusters for the purpose of their improvement. Hierarchical clustering is used in video clustering because it is easy to handle the similarity of extracted features from video, and it can represent the depth and granularity by the level of tree (Okamoto, Yasugi, Babaguchi, & Kitahashi, 2002).

Video Classification

While clustering is an unsupervised learning method, classification is a way to categorize or assign class labels to a pattern set under the supervision. Decision boundaries are generated to discriminate between patterns belonging to different classes. The data set is initially partitioned into training and test sets, and the classifier is trained on the former. A framework to enable semantic video classification and indexing in a specific video domain (medical video) was proposed (Fan, Luo, & Lin, 2003). VideoGraph, a tool for finding similar scenes in a video, was studied (Pan & Faloutsos, 2001). A method for classification of different kinds of videos that uses the output of a concise video summarization technique that forms a list of key frames was presented (Lu, Drew, & Au, 2001).

Pattern Finding

A general framework for video data mining was proposed to address the issue of how to extract previously unknown knowledge and detect interesting patterns (Oh, Lee, Kote, & Bandi, 2003). In the work, they develop how to segment the incoming raw video stream into meaningful pieces, and how to extract and represent some feature (i.e., motion) for characterizing the segmented pieces. Then, the motion in a video sequence is expressed as an accumulation of quantized pixel differences among all frames in the video segment. As a result, the accumulated motions of the segment are represented as a two dimensional matrix. Further, a method to capture the location of motions occurring in a segment is developed using the same matrix. How to cluster those segmented pieces using the features (the amount and the location of motion) extracted by the matrix above is studied. Also, an algorithm is investigated to determine whether a segment has normal or abnormal events by clustering and modeling normal events, which occur most frequently. In addition to deciding normal or abnormal, the algorithm computes a Degree of Abnormality of a segment, which represents the distance of a segment from the existing segments in relation to normal events.

A fast dominant motion extraction scheme called Integral Template Match (ITM), and a set of qualitative and quantitative description schemes were proposed (Lan, Ma, & Zhang, 2003). A video database management framework and its strategies for video content structure and events mining were introduced (Zhu, Aref, Fan, Catlin, & Elmagarmid, 2003). The methods of extracting editing rules from video stream were proposed by introducing a new data mining technique (Matsuo, Amano, & Uehara, 2002).

FUTURE TRENDS

As mentioned above, there have been a number of attempts to apply clustering methods to video data. However, these classical clustering techniques only create clusters but do not explain why a cluster has been established (Perner, 2002). The conceptual clustering method builds clusters and explains why a set of objects confirms a cluster. Thus, conceptual clustering is a type of learning by observations, and it is a way of summarizing data in an understandable manner. In contrast to hierarchical clustering methods, conceptual clustering methods build the classification hierarchy based on merging two groups. The algorithm properties are flexible in order to dynamically fit the hierarchy to the data. This allows incremental incorporation of new instances into the existing hierarchy and updating this hierarchy according to the new instance. A concept hierarchy is a directed graph in which the root node represents the set of all input instances and the terminal nodes represent individual instances. Internal nodes stand for the sets of instances attached to them and represent a super-concept. The super-concept can be represented by a generalized representation of this set of instances such as the prototype, the method or a user-selected instance. We can find a work applying this conceptual mining to image domain (Perner, 1998). However, we cannot find any papers related to conceptual clustering for video data. Since it is important to

understand what a created cluster means semantically, we need to study how to apply conceptual clustering to video data.

In fact, one of the most important techniques for data mining is association-rule mining since it is most efficient way to find unknown patterns and knowledge. Therefore, we need to investigate how to apply association-rule mining to video data. For association-rule mining, we need a set of transactions (*D*), a set of the literals (or items, *I*), and an itemset (*X*) (Zhang & Zhang, 2002). After video is segmented into the basic units such as shots, scenes, and events, each segmented unit can be modeled as a transaction, and the features from a unit can be considered as the items contained in the transaction. In this way, video association mining can be transformed into problems of association mining in traditional transactional databases. A work using some associations among video shots to create a video summary was proposed (Zhu & Wu, 2003). But they did not come up with the concepts of transaction and itemset. We need to further investigate the optimal unit for the concept of transaction, and the possible items in a transaction of video to characterize it.

Also, we need to study whether video can be considered as time-series data. It looks positive since the behavior of a time-series data item is very similar to that of video data. A time-series data item has a value at any given time, and the value is changing over time. Similarly a feature of video has a value at any given time, and the value is changing over time. If video can be considered as time-series data, we can get the advantages of the techniques already developed for time-series data mining. When the similarity between data items is computed, the ordinary distance metrics, such as Euclidean distance, may not be suitable, because of its high dimensionality and time factor. In order to address this problem, alternative ways are used to get a more accurate measure of similarity; for example, Dynamic Time Warping and Longest Common Subsequences.

Although most of data mining techniques are in batch processing, including video data mining as well as conventional data mining, some applications need to be processed in real time or near real time. For example, the anomaly detection system in a surveillance video using data mining should be processed in real time. Therefore, we also need to examine online video data mining.

CONCLUSION

V

Data mining describes a class of applications that look for hidden knowledge or patterns in large amounts of data. Most of data mining research has been dedicated to alpha-numeric databases, and relatively less work has been done for the multimedia data mining. The current status and the challenges of video data mining which is a very premature field of multimedia data mining, are discussed in this paper. The issues discussed should be dealt with in order to obtain valuable information from vast amounts of video data.

REFERENCES

Aref, W., Hammad, M., Catlin, A.C., Ilyas, I., Ghanem, T., Elmagarmid, A., & Marzouk, M. (2003). Video query processing in the VDBMS testbed for video database research. *Proceeding of 1st ACM International Workshop on Multimedia Database* (pp. 25-32).

Fan, J., Luo, H., & Lin, X. (2003). Semantic video classification by integrating flexible mixture model with adaptive EM algorithm. *Proceedings of the 5th ACM SIGMM International Workshop on Multimedia Information Retrieval* (pp. 9-16).

Frey, B.J., & Jojic, N. (2003). Transformation-invariant clustering using the EM algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *25*(1), 1-17.

Goh, K.S., Chang, E., & Cheng, K.T. (2001). SVM binary classifier ensembles for image classification. *Proceedings of the 10th International Conference on Information and Knowledge Management* (pp. 395-402).

Hsu, W., Dai, J., & Lee, M.L. (2003). Mining viewpoint patterns in image databases. *Proceeding of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 553-558).

Lan, D.J., Ma, Y.F., & Zhang, H.J. (2003). A novel motion-based representation for video mining. *Proceedings of 2003 IEEE International Conference on Multimedia and Expo, (ICME '03), Vol.3* (pp. 469-472).

Li, T., Li, Q., Zhu, S., & Ogihara, M. (2002). A survey on wavelet applications in data mining. *ACM SIGKDD Explorations Newsletter*, *4*(2), 49-68.

Lu, C., Drew, M.S., & Au, J. (2001). Classification of summarized videos using hidden Markov models on compressed chromaticity signatures. *Proceeding of 9th ACM International Conference on Multimedia* (pp. 479-482).

Lu, H., & Tan, Y.P. (2002). On model-based clustering of video scenes using scenelets. *Proceedings of 2002 IEEE International Conference on Multimedia and Expo, Vol.1* (pp. 301-304).

Matsuo, Y., Amano, M., & Uehara, K. (2002). Mining video editing rules in video streams. *Proceeding of the 10th ACM International Conference on Multimedia* (pp. 255-258).

Mitra, S., & Acharya, T. (2003). *Data mining: Multimedia, soft computing, and bioinformatics*. John Wiley & Sons, Inc.

Müller, H., & Pun, T. (2004). Learning from user behavior in image retrieval: Application of market basket analysis. *International Journal of Computer Vision, 56*(1/2), 65-77.

Ngo, C.W., Pong, T.C., & Zhang, H.J. (2001). On clustering and retrieval of video shots. *Proceeding of 9th ACM International Conference on Multimedia* (pp. 51-60).

Oh, J., Lee, J., Kote, S., & Bandi, B. (2003). Multimedia data mining framework for raw video sequences. *Mining Multimedia and Complex Data, Lecture Notes in Artificial Intelligence, Vol. 2797* (pp. 18-35). Springer Verlag.

Okamoto, H., Yasugi, Y., Babaguchi, N., & Kitahashi, T. (2002). Video clustering using spatio-temporal image with fixed length. *Proceedings of 2002 IEEE International Conference on Multimedia and Expo, Vol. 1* (pp. 53-56).

Pan, J.U., & Faloutsos, C. (2001). VideoGraph: A new tool for video mining and classification. *Proceedings of the 1st ACM/IEEE-CS Joint Conference in Digital Libraries* (pp. 116-117).

Perner, P. (1998). Using CBR learning for the low-level and high-level unit of a image interpretation system. In S. Singh (Ed.), *International Conference on Advances Pattern Recognition ICAPR98* (pp. 45-54). London: Springer Verlag.

Perner, P. (2002). *Data mining on multimedia data.* Springer.

Simoff, S.J., Djeraba, C., & Zaïane, O.R. (2002). MDM/KDD2002: Multimedia data mining between promises and problems. *ACM SIGKDD Explorations Newsletter, 4*(2), 118-121.

Thuraisingham, B., Clifton, C., Maurer, J., & Ceruti, M.G. (2001). Real-Time data mining of multimedia objects. *Proceedings of 4th IEEE International Symposium on Object-Oriented Real-Time distributed Computing, ISORC-2001* (pp. 360-365).

Yanai, K. (2003). Managing images: Generic image classification using visual knowledge on the Web. *Proceeding of the 11th ACM International Conference on Multimedia* (pp. 167-176).

Zaïane, O.R., Han, J., & Zhu, H. (2000). Mining recurrent Items in multimedia with progressive resolution refinement. *Proceedings of 16th International Conference on Data Engineering* (pp. 461-470).

Zhang, C., & Zhang, S. (2002). *Association rule mining*. Springer.

Zhu, X., Aref, W.G., Fan, J., Catlin, A.C., & Elmagarmid, A.K. (2003). Medical video mining for efficient database indexing, management and access. *Proceedings of the 19th International Conference on Data Engineering (ICDE '03)* (pp. 569-580).

Zhu, X., & Wu, X. (2003). Sequential association mining for video summarization. *Proceedings of 2003 IEEE International Conference on Multimedia and Expo, (ICME '03), Vol. 3* (pp. 333-336).

KEY TERMS

Classification: A method of categorizing or assigning class labels to a pattern set under the supervision.

Clustering: A process of mapping a data item into one of several clusters, where clusters are natural groupings for data items based on similarity metrics or probability density models.

Concept Hierarchy: A directed graph in which the root node represents the set of all input instances and the terminal nodes represent individual instances.

Conceptual Clustering: A type of learning by observations and a way of summarizing data in an understandable manner.

Degree of Abnormality: A probability that represents to what extent a segment is distant to the existing segments in relation with normal events.

Dendogram: It is a 'tree-like' diagram that summaries the process of clustering. Similar cases are joined by links whose position in the diagram is determined by the level of similarity between the cases.

Digital Multimedia: The bits that represent texts, images, audios, and videos, and are treated as data by computer programs.

Image Data Mining: A process of finding unusual patterns, and making associations between different images from large image databases. One could mine for associations between images, cluster images, classify images, as well as detect unusual patterns.

Image Processing: A research area analyzing and manipulating digital images for image enhancement, data compression, or pattern discovery.

Video Data Mining: A process of finding correlations and patterns previously unknown from large video databases.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 1185-1189, copyright 2005 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

View Selection in DW and OLAP: A Theoretical Review

Alfredo Cuzzocrea
University of Calabria, Italy

INTRODUCTION

Data Warehousing (DW) systems store *materialized views*, *data marts* and *data cubes*, and provide nicely data exploration and analysis interfaces via *OnLine Analytical Processing* (OLAP) (Gray et al., 1997) and *Data Mining* (DM) tools and algorithms. Also, *OnLine Analytical Mining* (OLAM) (Han, 1997) integrates the previous *knowledge discovery* methodologies and offers a meaningfully convergence between OLAP and DM, thus contributing to significantly augment the power of data exploration and analysis capabilities of knowledge workers. At the storage layer, the mentioned knowledge discovery methodologies share the problem of efficiently accessing, querying and processing multidimensional data, which in turn heavily affect the performance of knowledge discovery processes at the application layer. Due to the fact that OLAP and OLAM directly process data cubes/marts, and DM is more and more encompassing methodologies that are interested to multidimensional data, the problem of *efficiently representing data cubes by means of a meaningfully selected view set* is become of relevant interest for the Data Warehousing and OLAP research community.

This problem is directly related to the analogous problem of *efficiently computing the data cube* from a given relational data source (Harinarayan et al., 1996; Agarwal et al., 1996; Sarawagi et al., 1996; Zhao et al., 1997). Given a *relational data source* $\mathcal{R}$ and a target *data cube schema* $\mathcal{W}$, the *view selection problem* in OLAP deals with how to select and materialize views from $\mathcal{R}$ in order to compute the data cube $\mathcal{A}$ defined by the schema W by optimizing both the *query processing time*, denoted by $\mathcal{TQ}$, which models the amount of time required to answer a reference query-workload on the materialized view set, and the *view maintenance time*, denoted by $\mathcal{TM}$, which models the amount of time required to maintain the materialized view set when updates occur, under a given set of constraints $\mathcal{I}$ that, without any loss of generality, can be represented by a *space bound constraint* $\mathcal{B}$ limiting the overall occupancy of the views to be materialized (i.e., $\mathcal{I} = \langle \mathcal{B} \rangle$). It has been demonstrated (Gupta, 1997; Gupta & Mumick, 2005) that this problem is *NP-hard*, thus *heuristic schemes* are necessary. Heuristics are, in turn, implemented in the vest of *greedy algorithms* (Yang et al., 1997; Kalnis et al., 2002).

In this article, we focus the attention on state-of-the-art methods for the view selection problem in Data Warehousing and OLAP, and complete our analytical contribution with a theoretical analysis of these proposals under different selected properties that nicely model spatial and temporal complexity aspects of the investigated problem.

BACKGROUND

Before going into details, in this Section we provide the conceptual basis of our work, which is mainly related to the representation of multidimensional data cubes according to the ROLAP storage model. Let $\mathcal{R} = \langle \{R_0, R_1, \ldots, R_{M-1}\}, S \rangle$ be a relational data source, such that (*i*) R_i, with $0 \leq i \leq M-1$, is a relational table of form $R_i(A_{i,0}, A_{i,1}, \ldots, A_{i,|Ri|-1})$, such that $A_{i,j}$, with $0 \leq j \leq |R_i|-1$, is an *attribute* of R_i, and (*ii*) S it the *relational schema* that models associations among relational tables in $\{R_0, R_1, \ldots, R_{M-1}\}$. Let $\mathcal{W}$ be the goal data cube schema, which, without any loss of generality, can alternatively be modeled as a *star schema*, where a central *fact table* $\mathcal{F}$ is connected to multiple *dimensional tables* $\mathcal{T}_j$, or a *snowflake schema*, where dimensional tables are also *normalized* across multiple tables (Gray et al., 1997). At the conceptual level, the goal data cube A can also be modeled as a tuple $\mathcal{A} = \langle \mathcal{D}, \mathcal{H}, \mathcal{M} \rangle$, such that: (*i*) $\mathcal{D}$ is the set of *dimensions* of $\mathcal{A}$, (*ii*) $\mathcal{H}$ is the set of *hierarchies* associated to dimensions of $\mathcal{A}$, and (*iii*) $\mathcal{M}$ is the set of *measures* of $\mathcal{A}$. Dimensions model the *perspective of analysis* of the actual OLAP model. Hierarchies are hierarchical structures (e.g., *trees*) that capture hierarchical relationships among attributes of

dimensional tables. Measures model the *analysis goals* of the actual OLAP model.

Given an N-dimensional data cube $\mathcal{A}$, and the set of dimensions $\mathcal{D} = \{d_0, d_1, \ldots, d_{N-1}\}$ of $\mathcal{A}$, all the possible (simultaneous) combinations of sub-sets of $\mathcal{D}$ define a *lattice of cuboids* (Harinarayan et al., 1996), i.e. a set of hierarchically-organized multidimensional partitions of $\mathcal{A}$. Since real-life data cube have a large number of dimensions, the resulting number of cuboids N_C is large as well (Harinarayan et al., 1996). More precisely, this number is given by the following formula: $N_C = \prod_{k=0}^{N-1}\left(2^{L_k}+1\right)$, such that L_k denotes the depth of the hierarchy H_k associated to the dimension d_k, and the unary contribution is due to the aggregation ALL. Although N_C can became prohibitively large, the cuboid lattice offers several optimization opportunities for both the two complementary problems of computing the data cube (Harinarayan et al., 1996; Agarwal et al., 1996; Sarawagi et al., 1996; Zhao et al., 1997) and selecting the views to be materialized in order to efficiently representing the data cube.

VIEW SELECTION TECHNIQUES FOR DATA WAREHOUSING AND OLAP: THE STATE-OF-THE-ART

View selection is very related to the problem of computing the data cube (Harinarayan et al., 1996; Agarwal et al., 1996; Sarawagi et al., 1996; Zhao et al., 1997), as before to materialize data cube cells, views must be selected depending on spatial and (query) time constraints. Harynarayan *et al.* (1996) first consider the problem of efficiently computing a data cube starting from the relational source. The goal is that of optimizing the query processing time, under a given space bound. To this end, Harynarayan *et al.* (1996) develop a greedy algorithm working on the cuboid lattice that tries to optimize the so-called BPUS (*Benefit Per Unit Space*) that fine-grainy models the spatial cost needed to represent materialized views. Under an optimization-oriented view of the problem, this algorithm traverses the cuboid lattice and, at each step, materializes those cuboids that, overall, give the *greatest benefit* towards improving the query processing time while lowering the BPUS. Gupta (1997) first improves this methodology with the aim of specializing it towards the proper data cube model. The result consists in introducing an elegant graph-based notion to reason on views and their underlying base relations, called AND-/OR-DAG (*Direct Acyclic Graph*). An AND-/OR-DAG is a graph such that (*i*) leafs nodes represent the base relations stored in the relational data source, (*ii*) the root represents the view to be materialized, and (*iii*) internal nodes are classified in two classes, namely AND nodes and OR nodes. AND nodes model relational algebra operations like join, projection and selection, which apply on base relations or their combinations that, in turn, model operators. OR nodes model a set of *equivalence expressions* (in terms of SQL statements generating equivalent views and intermediate views) that can be alternatively used when a choice must be done. Based on this nice theoretical model, Gupta (1997) proposes a greedy algorithm that inspects the AND-/OR-DAG in order to determine the final view set via optimizing both query processing time and view maintenance time, under a given space bound. With respect to the previous work of Harynarayan *et al.* (1996), the most significant novelty carried out by (Gupta, 1997) is represented by the amenity of considering as parameter to be optimized the view maintenance time, beyond the query processing time. Gupta and Mumick further consolidate the theory of the view selection problem with maintenance time constraint in (Gupta & Mumick, 1999), where the *maintenance-cost view-selection problem* is formalized as an extension of the baseline view selection problem. All these research results have then been synthesized in (Gupta & Mumick, 2005).

Yang *et al.* (1997) propose a set of algorithms for the view selection problem that have the particular characteristic of finding, among all the possible ones, the sub-optimal solution capable of obtaining the best *combined benefit* between two critical factors, namely the *maximization* of query performance and the *minimization* of maintenance cost. Similarly to other proposals, the main idea of this approach consists in analyzing typical queries in order to detect *common intermediate results* that can be shared among queries with the aim of reducing computational overheads, thus improving the performance.

Agrawal *et al.* (2001) demonstrate the effectiveness and the reliability of view selection tools within commercial Data Warehousing and OLAP servers in the context of the *AutoAdmin* project (Agrawal et al., 2000; Agrawal et al., 2006). In more detail, (Agrawal et al., 2000; Agrawal et al., 2006) describe algorithms (implemented within the core layer of the related tool

called *AutoAdmin*) that allow us to select the optimal view set under a complex metrics involving query cost, maintenance cost and index construction cost. *AutoAdmin* is embedded within the well-know DBMS *Microsoft SQL Server 2000*.

DynaMat is a dynamic view management system proposed by Kotidis and Rossopulos (2001). The main idea underlying *DynaMat* is that of modeling the view selection problem in terms of a traditional *memory management problem*, which is typical of operating systems research. In *DynaMat*, a *pool* of materialized views is maintained in dependence on query monitoring reports and gathered statistics. Views are added to the pool on the basis of the frequency they are involved by input queries, or removed if they are no longer involved. Strategies for removing victim views, inspired from similar methods of operating systems, are presented and evaluated (Kotidis & Rossopulos, 2001).

Lee *et al.* (2001; 2007) propose a very interesting optimization to the view maintenance problem that can be reasonably regarded as a valuable contribution over state-of-the-art research. The resulting approach is called *Delta Propagation Strategy*. The underlying method deals with the issue of efficiently maintaining the materialized view set in the presence of updates that can occur in the reference relational source. The main intuition here is noticing that the performance of any view maintenance system is heavily affected by the amount of I/O disk accesses to the base relations from which views are derived (e.g., based on an AND-/OR-DAG). Inspired by this main intuition, Lee *et al.* (2001; 2007) rigorously model the cost needed to compute a *changed* materialized view V, denoted by ΔV, in dependence of updates that occur in the base relations R_i, being changed relations denoted by ΔR_i. Then, they introduce the so-called *delta expressions*, which are maintenance expressions derived from formulas and equivalences coming from *Relational Algebra*. Delta expressions allow us to minimize the cost of computing ΔV via reducing the overall number of I/O disk accesses to (changed) base relations. Finally, since computing optimal delta expressions from a given set of changed relations is still a combinatory problem, the so-called *delta propagation tree* is introduced, along with a *dynamic programming algorithm* that exploits the hierarchical model offered by this structure in order to efficiently accomplish the view maintenance task. The same approach has been then targeted to data cubes by the same authors (Lee & Kim, 2006) via introducing the concept of *delta cuboid* that, similarly to the relational case, effectively captures changes that can happen in a cuboid.

Mistry *et al.* (2001) attack the view selection problem under a different perspective of research, i.e. the issue of materializing *transient* views in Data Warehousing and OLAP systems. Transient views refer to views that are temporarily materialized, e.g. during the evaluation of complex OLAP queries against very large Data Warehouses, and are different from the materialized views considered until now, which should be more precisely referred as *permanent* views, i.e. views that are materialized permanently. The issue of materializing transient views is also known in literature under the term: *multi-query optimization problem* (Sellis, 1988). The novelty of the Mistry *et al.*'s approach (2001) relies in the fact that an integrated methodology taking into consideration the selection of both transient and permanent views in a combined manner is proposed. Indeed, this aspect can be reasonably intended as innovative with respect to previous research experiences. In this direction, Mistry *et al.* (2001) devise a quite complex methodology that encompasses the following aspects: (*i*) temporarily materialization of common sub-expressions in order to reduce the overall cost of different maintenance plans, (*ii*) selection of auxiliary expressions that further improve the overall view maintenance task, (*iii*) selection of the optimal maintenance plan (which can alternatively be incremental or with re-computation from the scratch) for each selected view, (*iv*) minimization of the overall maintenance cost by meaningfully combining all the previous aspects. The reasoning unit of the Mistry *et al.*'s approach (2001) approach still is the AND-/OR-DAG, and a greedy algorithm is again exploited to transient and permanent view selection and materialization purposes.

Kalnis *et al.* (2002) propose a novel perspective to face-off the annoying view selection and materialization problem. Starting from specific requirements of real-life Data Warehousing and OLAP systems, such as the need of analyzing multidimensional data according to numerous perspectives of analysis, they propose the application of *randomized local search algorithms*, and test the efficiency of these algorithms against several synthetic data sets and different query distributions shaping high-heterogeneous query-workloads that can be posed to the system. In (Kalnis et al., 2002), authors correctly assert that, for low-dimensional data cubes, state-of-the-art proposals are able to provide

efficiency within satisfactory space and maintenance time constraints, whereas when data cubes grow in dimension number and size this efficiency drastically decreases. In this context, and the application of randomized approaches seems to be the most pertinent one. Authors model the overall search space of the underlying optimization problem in terms of an *undirected graph* where: (*i*) nodes represent *candidate solutions* (i.e., set of views for the particular selection and materialization instance); (*ii*) arcs represent simple transformations among set of views activated by means of transactions, and capture the hierarchical relationships among such sets; (*iii*) a cost, depending on both the space and maintenance time constraints, is associated to each node, and the final goal is that of finding the solution that minimizes this cost. Authors also review state-of-the-art randomized algorithms, and develop a very comprehensive experimental study that shows the different trade-offs in terms of query performance and maintenance cost of classical methods and other more focused approaches such as (Gupta, 1997; Gupta & Mumick, 2005; Shukla et al., 1998), and also *EX*, a *branch-&-bound* algorithm that exhaustively searches the overall view space in order to avoid unnecessary checks thus speeding-up the execution time.

A *wavelet-based framework* for making flexible data cube views for OLAP is proposed by Smith *et al.* (2004). Wavelets have been extensively used in OLAP to obtain data cube compression in the context of *approximate query answering techniques* (Vitter et al., 1998; Chakrabarti et al., 2000). The framework proposed by Smith *et al.* (2004) asserts to decompose the input data cube into a set of the so-called *wavelet view elements*, which are obtained by means of *partial SQL aggregations* over the input data cube. Depending on the way of generating wavelet view elements, they can be of *aggregated*, *intermediate* or *residual* kinds. The main idea of this approach consists in adopting the well-known *wavelet decomposition hierarchy* (Stollnitz et al., 1996) to represent the dependencies among view elements instead of the widely-adopted view dependency hierarchy defined by the cuboid lattice (Harinarayan et al., 1996). This is meant to achieve more flexibility in several OLAP-oriented activities such as querying and analyzing data cubes. In this vest, the proposed wavelet-based framework can also be used for the view selection problem, as authors state in (Smith et al., 2004). According to this framework, an input data cube is represented by means of a view element set that satisfies the following properties: (*i*) *perfect reconstruction*, which asserts that the entire data cube can be fully-reconstructed from the view element set; (*ii*) *non-expansiveness*, which asserts that the generation of child view elements from ancestor ones does not augment the volume of the data cube; (*iii*) *distributivity*, which asserts that view elements can be computed from other ones in the wavelet decomposition hierarchy; (*iv*) *separability*, which asserts that a multidimensional view element can be obtained by applying partial SQL aggregations over each single dimension of the target data cube separately. Similarly to other approaches, wavelet view elements are handled by means of a hierarchical data structure, called *View Element Graph* (VEG). On top of VEG, authors propose two different greedy algorithms for the view selection problem. The first one is based on the assumption claiming that views of the view element set compose *a complete and non-redundant basis of the data cube*. The second one relaxes the previous assumption, and finds the optimal view set that minimizes the average throughput cost with respect to the spatial cost, i.e. the cost needed to represent the materialized views in secondary memory.

THEORETICAL ANALYSIS AND RESULTS

In this Section, we provide the theoretical analysis and related results of those techniques that we retain as the most "representative" ones among the state-of-the-art proposals. In our theoretical analysis, we introduce the following parameters:

- *Spatial Complexity*, which models the spatial complexity of a data cube obtained in terms of the total amount of tuples (in the case of the ROLAP storage organization) or cells (in the case of the MOLAP storage organization) that must be materialized to represent the cube.
- *Time Complexity*, which models the temporal complexity required to find the optimal view set to be materialized.
- *Update Complexity*, which models the temporal complexity of updating a data cube obtained in terms of the total amount of tuples (for ROLAP)

or cells (for MOLAP) that must be updated as the consequence of load or refresh operations at the relational data source.

Table 1 summarizes theoretical results for some representative state-of-the-art view selection and materialization techniques.

Note that for *all* the considered view selection and materialization techniques, the spatial complexity is obviously the same, as it only depends on the particular OLAP storage model used to represent the final data cube. In particular, for an N-dimensional data cube, if the ROLAP model is used, this complexity is $O\left(\sum_{j=0}^{P-1} |R_j|\right)$, such that $|R_j|$ denotes the cardinality of the relation R_j (i.e., the number of tuples of R_j). Explicitly note that P is such that $P \geq N$, as in a ROLAP schema dimensions can be mapped in single or normalized dimensional tables. Otherwise, if the MOLAP model is instead used, this complexity is $O\left(\prod_{k=0}^{N-1} |d_k|\right)$, such that $|d_k|$ denotes the cardinality of the dimension d_k (i.e., the number of members of d_k).

For what regards (Gupta, 1997; Gupta & Mumick, 2005), the time complexity of finding the optimal view set on the AND-/OR-DAG is $O(k \times n^2)$, such that k is the number of stages required by the greedy algorithm, and n is the number of nodes of the AND-/OR-DAG. The update complexity is instead exponential in the number of nodes of the AND-/OR-DAG.

Table 1. Theoretical analysis of some representative view selection and materialization techniques

Technique	Spatial Complexity		Time Complexity	Update Complexity
(Gupta, 1997; Gupta & Mumick, 2005)	ROLAP	$O\left(\sum_{j=0}^{P-1} \lvert R_j \rvert\right)$	$O(k \times n^2)$	EXPTIME(n)
	MOLAP	$O\left(\prod_{k=0}^{N-1} \lvert d_k \rvert\right)$		
(Lee et al., 2001; Lee et al., 2007)	ROLAP	$O\left(\sum_{j=0}^{P-1} \lvert R_j \rvert\right)$	$O(2^n \times n)$	EXPTIME(n)
	MOLAP	$O\left(\prod_{k=0}^{N-1} \lvert d_k \rvert\right)$		
(Mistry et al., 2001)	ROLAP	$O\left(\sum_{j=0}^{P-1} \lvert R_j \rvert\right)$	$O(k \times n^2)$	EXPTIME(n)
	MOLAP	$O\left(\prod_{k=0}^{N-1} \lvert d_k \rvert\right)$		
(Smith et al., 2004)	ROLAP	$O\left(\sum_{j=0}^{P-1} \lvert R_j \rvert\right)$	$O\left((N+1) \times \prod_{k=0}^{N-1} (2 \times \lvert d_k \rvert - 1)\right)$	EXPTIME(n)
	MOLAP	$O\left(\prod_{k=0}^{N-1} \lvert d_k \rvert\right)$		

For what regards (Lee et al., 2001; Lee et al., 2007), the time complexity of finding the optimal delta expression on the delta propagation tree is $O(2^n \times n)$, such that n is the number of base relations from which views are derived. The update complexity is instead exponential in the number of base relations.

For what regards (Mistry et al., 2001), the time complexity is the same of (Gupta, 1997; Gupta & Mumick, 2005), i.e. $O(2^n \times n)$, as the AND-/OR-DAG is still used as baseline computational model. Indeed, (Mistry et al., 2001) mostly focuses on the problem of maintaining the already-materialized view set. While the update complexity is still exponential in the number of nodes of the AND-/OR-DAG, Mistry *et al.* (2001) develop very useful closed formulas for estimating the maintenance cost of a materialized view set (see the paper for further details).

Finally, for what regards (Smith et al., 2004) the time complexity of extracting the non-redundant basis of the data cube modeling the optimal view set (i.e., the first of the two proposed view selection algorithms) is $O\left((N+1)\times\prod_{k=0}^{N-1}(2\times|d_k|-1)\right)$, such that N is the number of dimensions of the data cube and $|d_k|$ is the cardinality of the dimension d_k. The update complexity is instead exponential in the number of nodes of the VEG.

FUTURE TRENDS

Although of traditional nature, the problem of view selection and materialization in Data Warehousing and OLAP systems still plays a significant role today. Mostly, the research effort should be oriented towards novel applications contexts where this problem demands for specialized solutions, beyond the capabilities of state-of-the-art techniques. Some of these contexts are: (*i*) *high-dimensional data cubes*, (*ii*) *XML native database systems*, (*iii*) *sensor database systems*, (*iv*) *RFID database systems*, (*v*) *trajectory database systems*, (*vi*) *incomplete information bases*.

CONCLUSION

This article has presented and discussed in detail a comprehensive number of view selection and materialization techniques for Data Warehousing and OLAP systems, by also putting in evidence related benefits and limitations. Then, some of these techniques have been selected as the most representative ones, as fundamental issues of the investigated problem have been proposed just in these papers. For these techniques, a theoretical analysis under different parameters that nicely model spatial and temporal complexity aspects has completed the overall contribution of the article.

REFERENCES

Agarwal, S., Agrawal, R., Deshpande, P.M., Gupta, A., Naughton, J.F., Ramakrishnan, R., & Sarawagi, S. (1996). On the Computation of Multidimensional Aggregates. *Proceedings of the 22nd International Conference on Very Large Data Bases*, 506-521.

Agrawal, S., Bruno, N., Chaudhuri, S., & Narasayya, V.R. (2006). AutoAdmin: Self-Tuning Database Systems Technology. *IEEE Data Engineering Bulletin*, 29(3), 7-15.

Agrawal, S., Chaudhuri, S., & Narasayya, V.R. (2000): Automated Selection of Materialized Views and Indexes in SQL Databases. *Proceedings of the 26th International Conference on Very Large Data Bases*, 496-505.

Agrawal, S., Chaudhuri, S., & Narasayya, V.R. (2001). Materialized View and Index Selection Tool for Microsoft SQL Server 2000. *Proceedings of the 2001 ACM International Conference on Management of Data*, 608.

Chakrabarti, K., Garofalakis, M., Rastogi, R., & Shim, K. (2000). Approximate Query Processing using Wavelets. *Proceedings of the 26th International Conference on Very Large Data Bases*, 111-122.

Gray, J., Chaudhuri, S., Bosworth, A., Layman, A., Reichart, D., & Venkatrao, M. (1997). Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals. *Data Mining and Knowledge Discovery*, 1(1), 29-53.

Gupta, H. (1997). Selection of Views to Materialize in a Data Warehouse. *Proceedings of the 6th International Conference on Database Theory*, 98-112.

Gupta, H., & Mumick, I.S. (1999). Selection of Views to Materialize Under a Maintenance Cost Constraint. *Proceedings of the 8th International Conference on Database Theory*, 453-470.

Gupta, H., & Mumick, I.S. (2005). Selection of View to Materialize in a Data Warehouse. *IEEE Transactions on Knowledge and Data Engineering*, 17(1), 24-43.

Han, J. (1997). OLAP Mining: An Integration of OLAP with Data Mining. *Proceedings of 7th IFIP 2.6 Working Conference on Database Semantics*, 1-9.

Harinarayan, V., Rajaraman, A., & Ullman, J.D. (1996). Implementing Data Cubes Efficiently. *Proceedings of the 1996 ACM International Conference on Management of Data*, 205-216.

Kalnis, P., Mamoulis, N., & Papadias, D. (2002). View Selection using Randomized Search. *Data & Knowledge Engineering*, 42(1), 89-111.

Kotidis, Y., & Roussopoulos, N. (2001). A Case for Dynamic View Management. *ACM Transactions on Database Systems*, 26(4), 388-423.

Lee, K.Y., & Kim, M.-H. (2006). Efficient Incremental Maintenance of Data Cubes. *Proceedings of the 32nd International Conference on Very Large Data Bases*, 823-833.

Lee, K.Y., Son, J.H., & Kim, M.-H. (2001). Efficient Incremental View Maintenance in Data Warehouses. *Proceedings of the 10th ACM International Conference on Information and Knowledge Management*, 349-357.

Lee, K.Y., Son, J.H., & Kim, M.-H. (2007). Reducing the Cost of Accessing Relations in Incremental View Maintenance. *Decision Support Systems*, 43(2), 512-526.

Mistry, H., Roy, P., Sudarshan, S., & Ramamritham, K. (2001). Materialized View Selection and Maintenance Using Multi-Query Optimization. *Proceedings of the 1997 ACM International Conference on Management of Data*, 307-318.

Sarawagi, S., Agrawal, R., & Gupta, A. (1996). On Computing the Data Cube. *Technical Report, IBM Almaden Research Center*.

Sellis, T.K. (1988). Multiple-Query Optimization. *ACM Transactions on Database Systems*, 13(1), 23-52.

Shukla, A., Deshpande, P., & Naughton, J.F. (1998). Materialized View Selection for Multidimensional Datasets. *Proceedings of the 24th International Conference on Very Large Data Bases*, 488-499.

Smith, J.R., Li, C., and Jhingran, A. (2004). A Wavelet Framework for Adapting Data Cube Views for OLAP. *IEEE Transactions on Knowledge and Data Engineering*, 16(5), 552-565.

Stollnitz, E.J., Derose, T.D., & Salesin, D.H. (1996). *Wavelets for Computer Graphics*. Morgan Kauffmann Publishers.

Vitter, J.S., Wang, M., & Iyer, B. (1998). Data Cube Approximation and Histograms via Wavelets. *Proceeding of the 7th ACM International Conference on Information and Knowledge Management*, 96-104.

Yang, J., Karlapalem, K., & Li, Q. (1997). Algorithms for Materialized View Design in Data Warehousing Environment. *Proceedings of the 23rd International Conference on Very Large Data Bases*, 136-145.

Zhao, Y., Deshpande, P.M., & Naughton, J.F.: An Array-based Algorithm for Simultaneous Multidimensional Aggregates. *Proceedings of the 1997 ACM International Conference on Management of Data*, 159-170.

KEY TERMS

Cuboid: A multidimensional partition (or equally, sub-cube) of a data cube obtained by aggregating data along a given sub-set of the dimensions.

Cuboid Lattice: Hierarchical representation of the generation relationships among cuboids of a given data cube.

Data Cube: A multidimensional data structure storing OLAP data as measures (of interest) that are indexed in a multidimensional space defined by the dimensions of the cube.

Data Warehousing (DW): A methodology for integrating, representing and managing massive heterogeneous and distributed data sets.

Multidimensional View: A portion of a multidimensional database that stores data related to a specific subject of (OLAP) analysis.

On-Line Analytical Processing (OLAP): A methodology for exploring and querying massive DW data according to multidimensional and multi-resolution abstractions of them.

Relational View: A portion of a relational database that stores data related to a specific application-domain/process.

Visual Data Mining from Visualization to Visual Information Mining

Herna L. Viktor
University of Ottawa, Canada

Eric Paquet
National Research Council, Canada

INTRODUCTION

The current explosion of data and information, which are mainly caused by the continuous adoption of data warehouses and the extensive use of the Internet and its related technologies, has increased the urgent need for the development of techniques for intelligent data analysis. Data mining, which concerns the discovery and extraction of knowledge chunks from large data repositories, addresses this need. Data mining automates the discovery of hidden patterns and relationships that may not always be obvious. Data mining tools include classification techniques (such as decision trees, rule induction programs and neural networks) (Kou et al., 2007); clustering algorithms and association rule approaches, amongst others.

Data mining has been fruitfully used in many of domains, including marketing, medicine, finance, engineering and bioinformatics. There still are, however, a number of factors that militate against the widespread adoption and use of this new technology. This is mainly due to the fact that the results of many data mining techniques are often difficult to understand. For example, the results of a data mining effort producing 300 pages of rules will be difficult to analyze. The visual representation of the knowledge embedded in such rules will help to heighten the comprehensibility of the results. The visualization of the data itself, as well as the data mining process should go a long way towards increasing the user's understanding of and faith in the data mining process. That is, data and information visualization provide users with the ability to obtain new insights into the knowledge, as discovered from large repositories.

This paper describes a number of important visual data mining issues and introduces techniques employed to improve the understandability of the results of data mining. Firstly, the visualization of data prior to, and during, data mining is addressed. Through *data* visualization, the quality of the data can be assessed throughout the knowledge discovery process, which includes data preprocessing, data mining and reporting. We also discuss *information* visualization, i.e. how the knowledge, as discovered by a data mining tool, may be visualized throughout the data mining process. This aspect includes visualization of the results of data mining as well as the learning process. In addition, the paper shows how virtual reality and collaborative virtual environments may be used to obtain an immersive perspective of the data and the data mining process as well as how visual data mining can be used to directly mine functionality with specific applications in the emerging field of proteomics.

BACKGROUND

Human beings intuitively search for novel features, patterns, trends, outliers and relationships in data (Han and Kamber, 2006). Through visualizing the data and the concept descriptions obtained (e.g., in the form of rules), a qualitative overview of large and complex data sets can be obtained. In addition, data and rule visualization can assist in identifying regions of interest and appropriate parameters for more focused quantitative analysis. The user can thus get a "rough feeling" of the quality of the data, in terms of its correctness, adequacy, completeness, relevance, etc. The use of data and rule visualization thus greatly expands the range of models that can be understood by the user, thereby easing the so-called "accuracy versus understandability" tradeoff (Valdes and Barton, 2007).

Data mining techniques construct a model of the data through repetitive calculation to find statistically significant relationships within the data. However, the human visual perception system can detect patterns

within the data that are unknown to a data mining tool. This combination of the various strengths of the human visual system and data mining tools may subsequently lead to the discovery of novel insights and the improvement of the human's perspective of the problem at hand. Visual data mining harnesses the power of the human vision system, making it an effective tool to comprehend data distribution, patterns, clusters and outliers in data (Blanchard et al., 2007).

Visual data mining is currently an active area of research. Examples of related commercial data mining packages include the *MultiMediaMiner* data mining system, *See5* which forms part of the RuleQuest suite of data mining tools, *Clementine* developed by Integral Solutions Ltd (ISL), *Enterprise Miner* developed by SAS Institute, *Intelligent Miner* produced by IBM, and various other tools. Neural network tools such as *NeuroSolutions* and *SNNS* and Bayesian network tools including *Hugin*, *TETRAD*, and *Bayesware Discoverer*, also incorporates extensive visualization facilities. Examples of related research projects and visualization approaches include *MLC++*, *WEKA*, *AlgorithmMatrix*, *C4.5/See5* and NCBI GEO amongst others (Barret et al., 2007).

Visual data mining integrates data visualization and data mining and is closely related to computer graphics, multimedia systems, human computer interfaces, pattern recognition and high performance computing.

DATA AND INFORMATION VISUALIZATION

Data Visualization

Data visualization provides a powerful mechanism to aid the user during both data preprocessing and the actual data mining. Through the visualization of the original data, the user can browse to get a "feel" for the properties of that data. For example, large samples can be visualized and analyzed (Barret et al., 2007). In particular, visualization may be used for outlier detection, which highlights surprises in the data, i.e. data instances that do not comply with the general behavior or model of the data (Sun et al., 2007). In addition, the user is aided in selecting the appropriate data through a visual interface. Data transformation is an important data preprocessing step. During data transformation, visualizing the data can help the user to ensure the correctness of the transformation. That is, the user may determine whether the two views (original versus transformed) of the data are equivalent. Visualization may also be used to assist users when integrating data sources, assisting them to see relationships within the different formats.

Data visualization techniques are classified in respect of three aspects. Firstly, their focus, i.e. symbolic versus geometric; secondly their stimulus (2D versus 3D); and lastly, their display (static or dynamic). In addition, data in a data repository can be viewed as different levels of granularity or abstraction, or as different combinations of attributes or dimensions. The data can be presented in various visual formats, including box plots, scatter plots, 3D-cubes, data distribution charts, curves, volume visualization, surfaces or link graphs, amongst others (Gardia-Osorio and Fyfe, 2008).

For instance, 3D-cubes are used in relationship diagrams, where the data are compared as totals of different categories. In surface charts, the data points are visualized by drawing a line between them. The area defined by the line, together with the lower portion of the chart, is subsequently filled. Link or line graphs display the relationships between data points through fitting a connecting line (Guo et al., 2007). They are normally used for 2D data where the X value is not repeated.

Advanced visualization techniques may greatly expand the range of models that can be understood by domain experts, thereby easing the so-called accuracy-versus-understandability trade-off. However, due to the so-called "curse of dimensionality", which refers to the problems associated with working with numerous dimensions, highly accurate models are usually less understandable, and vice versa. In a data mining system, the aim of data visualization is to obtain an initial understanding of the data and the quality thereof. The actual accurate assessment of the data and the discovery of new knowledge are the tasks of the data mining tools. Therefore, the visual display should preferably be highly understandable, possibly at the cost of accuracy.

The use of one or more of the above-mentioned data visualization techniques thus helps the user to obtain an initial model of the data, in order to detect possible outliers and to obtain an intuitive assessment of the quality of the data used for data mining. The visualization of the data mining process and results is discussed next.

Information Visualization

It is crucial to be aware of what users require for exploring data sets, small and large. The driving force behind visualizing data mining models can be broken down into two key areas, namely understanding and trust. Understanding means more than just comprehension; it also involves context. If the user can understand what has been discovered in the context of the business issue, he will trust the data and the underlying model and thus put it to use. Visualizing a model also allows a user to discuss and explain the logic behind the model to others. In this way, the overall trust in the model increases and subsequent actions taken as a result are justifiable (Blanchard et al., 2007). Visualization thus aids us to determine whether the data mining process is of high economic utility, i.e. it is adding value especially when considering large-scale real-world data mining projects (Hilderman, 2006).

The art of information visualization can be seen as the combination of three well defined and understood disciplines, namely cognitive science, graphics art and information graphics. A number of important factors have to be kept in mind when visualizing both the execution of the data mining algorithm (process visualization), e.g. the construction of a decision tree, and displaying the results thereof (result visualization). The visualization approach should provide an easy understanding of the domain knowledge, explore visual parameters and produce useful outputs. Salient features should be encoded graphically and the interactive process should prove useful to the user (Jankun-Kelly et al., 2007).

The format of knowledge extracted during the mining process depends on the type of data mining task and its complexity. Examples include classification rules, association rules, temporal sequences, casual graphs and tail trees (Klemela, 2007). Visualization of these data mining results involves the presentation of the results or knowledge obtained from data mining in visual forms, such as decision trees, association rules, clusters, outliers and generalized rules (Hruschka et al., 2007). For example, the Silicon Graphics (SGI) MineSet (MLC++) and the Blue Martini Software toolsets use connectivity diagrams to visualize decision trees, and simple Bayesian and decision table classifiers (Erbacher and Teerling, 2006). Other examples include the Iris Explorer system that offers techniques ranging from simple graphs to multidimensional animation (NAG, 2007); and SpectraMiner, an interactive data mining and visualization software for single particle mass spectroscopy (Zelenyuk et. al., 2006).

Visual Data Mining and Virtual Reality

Three-dimensional visualization has the potential to show far more information than two-dimensional visualization, while retaining its simplicity. This visualization technique quickly reveals the quantity and relative strength of relationships between elements, helping to focus attention on important data entities and rules. It therefore aids both the data preprocessing and data mining processes.

Many techniques are available to visualize data in three dimensions. For example, it is very common to represent data by glyphs. A glyph can be defined as a three-dimensional object suitable for representing data or subsets of data. The object is chosen in order to facilitate both the visualization and the data mining process. The glyph must be self-explanatory and unambiguous.

Three-dimensional visualization can be made more efficient by the use of virtual reality (VR). A virtual environment (VE) is a three-dimensional environment characterized by the fact that it is immersive, interactive, illustrative and intuitive. The fact that the environment is immersive is of great importance in data mining. In traditional visualization, the human subject looks at the data from outside, while in a VR environment the user is part of the data world. This means that the user can utilize all his senses in order to navigate and understand the data. This also implies that the representation is more intuitive. VR is particularly well adapted to representing the scale and the topology of various sets of data. That becomes even more evident when stereo visualization is utilized, since stereo vision allows the analyst to have a real depth perception. This depth perception is important in order to estimate the relative distances and scales between the glyphs. Such estimation can be difficult without stereo vision if the scene does not correspond to the paradigms our brain is used to processing. In certain cases, the depth perception can be enhanced by the use of metaphors.

In addition, we mention some non standard visualization techniques; for example iconic displays, dense pixel displays and stacked displays (Keim, 2002). The iconic display maps the attributes of a multidimensional data set to the features or parameters of an iconic representation e.g. color icons, sick figure icons and star

icons. A dense pixel display maps each dimension value to a colored pixel and groups the pixels belonging to each dimensions into adjacent areas. Stack displays are utilized to represent data which have a hierarchical partition by embedding coordinate systems inside each other.

Irrespectively of the representation chosen, the data exploration is more efficient by making use of the interaction and distortion techniques. In the case of the interaction technique, one can modify in real time the visualization scheme. Also, we may combine, relate and group various visual elements in order to facilitate the exploration of the data. As of the case of the distortion technique, it is based on the concept of multiresolution: a subset of the data is shown at high resolution while others are shown at low level. Such a technique can be either manual or automatic and is called, accordingly, interactive or dynamic.

Collaborative Virtual Environments (CVEs) can be considered as a major breakthrough in data mining (Jamieson et al., 2007). By analogy, they can be considered as the equivalent of collaborative agents in visualization. Traditionally, one or more analysts perform visualization at a unique site. This operational model does not reflect the fact that many enterprises are distributed worldwide and so are their operations, data and specialists. It is consequently impossible for those enterprises to centralize all their data mining operations in a single center. Not only must they collaborate on the data mining process, which can be carried out automatically to a certain extent by distributed and collaborative agents, but they must also collaborate on the visualization and the visual data mining aspects.

FUTURE TRENDS

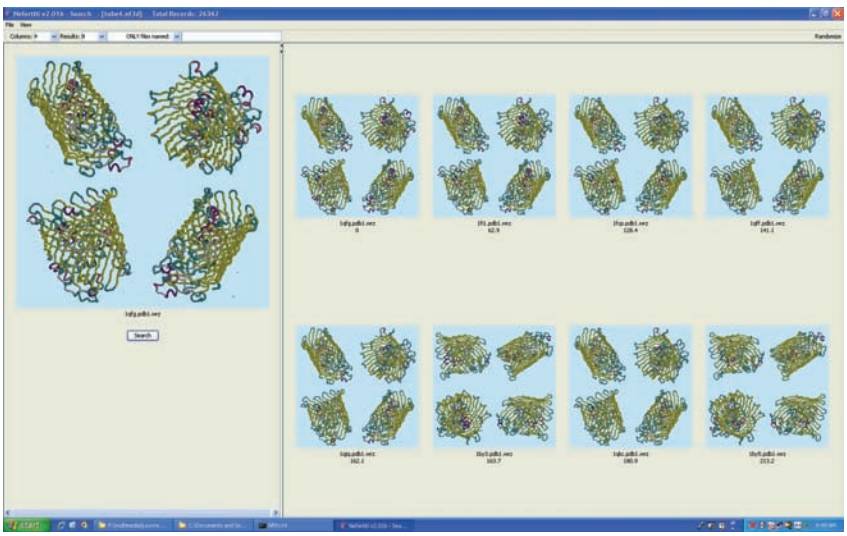
Visual Data Mining and Proteomics; when Visual Appearance Becomes Functionality

Visual data mining is not limited to data visualization per se. In some specific cases, the visual appearance of the data is related to their effective functionality and visual data mining becomes synonym of function mining; a highly attractive feature for many practical applications. In this section, such an approach is applied to the emerging field of proteomics and computer aided drug design.

Protein function analysis is an important research issue in molecular biology, bioinformatics and pharmaceutics. A protein's function is often dependent on the shape (visual appearance) and physical properties of the so-called active sites (local subparts) of the molecular surface. Current research suggests that, if two proteins have similar active sites, the function of the two proteins may be closely related. This observation is of importance for many reasons.

Consider a protein which has shown to be successful in a prescription drug developed to treat a

Figure 1.The CAPRI visual data mining system for proteins



terminal illness. However, this protein has serious contra-indications and causes severe adverse effects in a certain subset of the population. Suppose a protein with similar visual structure and functionality, but without these serious adverse effects, can be found. The subsequent modification of the harmful drug has obvious benefits.

For instance, the Content-based Analysis of Protein Structure for Retrieval and Indexing (CAPRI) visual data mining system addresses this issue (Paquet and Viktor, 2007). CAPRI is able to utilize the 3D structure of a protein, in order to find the *k* most similar structures.

The results against more than 26,000 protein structures as contained in the Protein Data Bank show that the system is able to accurately and efficiently retrieve similar protein structures with a very high precision-recall rate. For instance, figure 1 shows the retrieval of visually similar proteins based on their 3D shape. All proteins shown belong to the same family. Through the use of the CAPRI system, domain experts are able to find similar protein structures, using a "query by prototype" approach. In this way, they are aided in the task of labeling new structures effectively, finding the families of existing proteins, identifying mutations and unexpected evolutions.

The main benefit of 3D structural indexing is that the protein functionality is related to its 3D shape. 3D shape indexing is a natural way to index the functionality with all the foreseen applications in bioinformatics, genomic, as well as for the pharmaceutical industry.

CONCLUSION

The ability to visualize the results of a data mining effort aids the user to understand and trust the knowledge embedded in it. Data and information visualization provide the user with the ability to get an intuitive "feel" for the data and the results, e.g. in the form of rules, that is being created. This ability can be fruitfully used in many business areas, for example for fraud detection, diagnosis in medical domains and credit screening, amongst others.

Finally, the direct mining of visual information looks very promising in proteomics for the design of new drugs with fewer side effects.

REFERENCES

Barrett T., Troup D. B., Wilhite S. E., Ledoux P., Rudnev R., Evangelista C., Kim I. F., Soboleva A., Tomashevsky M. and Edgar R. (2007). NCBI GEO: Mining tens of millions of expression profiles - Database and tools update. *Nucleic Acids Research*, 35, D760-D765.

Blanchard J., Guillet F. and Briand H. (2006). Interactive visual exploration of association rules with rule focusing methodology. *Knowledge and Information Systems*, *13*(1), 43-75.

Erbacher, R. and Teerlink, S., (2006). Improving the computer forensic analysis process through visualization. *Communications of the ACM*, *49*(2), 71-75.

Garcia-Osorio C. and Fyfe C. (2008). Visualizing multi dimensional data. *Successes and New Directions in Data Mining*, (pp. 236-276). Hershey, PA: IGI Global.

Guo D., Liao K. and Morgan M. (2007). Visualizing patterns in a global terrorism incident database. *Environment and Planning, Planning and Design*, *34*(5), 767-784.

Han, J., & Kamber, M. (2006). *Data mining concepts and techniques*, 2nd Edition. San Francisco: Morgan Kaufmann.

Hilderman, R.J. (2006). Assessing the interestingness of discovered knowledge using a principled objective approach, *2nd SIGKDD Workshop on Utility-Based Data Mining* (Philadelphia, PA), 2006, 44-53.

Hruschka E. R. Jr., do Carmo N. M., de Oliveira V. A. and Bressan G. M. (2007). Markov-Blanket based strategy for translating a Bayesian classifier into a reduced set of classification rules. *HIS 2007* (Kaiserslautern, Germany), 192-197.

Jamieson R., Haffegee A., Ramsamy P. and Alexandrov V. (2007). Data forest: A collaborative version. *ICCS 2007* (Beijing, China), SLNCS 4488, 744-751.

Jankun-Kelly T. J., Kwan-Liu M. and Gertz M. (2007). A model and framework for visualization exploration. *Transactions on Visualization and Computer Graphics*, *12*(2), 357-369.

Keim D. A. (2002). Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Sciences*, *8*(1), 1-8.

Klemela J. (2007). Visualization of multivariate data with tail trees. *Information Visualization*, *6*(2), 109-122.

Kou G., Peng Y., Shi Y. and Chen Z. (2007). Epsilon support vector and large-scale data mining problems. *ICCS-2007* (Beijing, China), SLNCS 4489, 874-881.

Numerical Algorithms Group (2007). *Iris explorer*. http://www.nag.co.uk/welcome_iec.asp

Paquet E. and Viktor H. L. (2007). Exploring protein architecture using 3D shape-based signatures. *ECMB 2007* (Lyon, France), 1204-1208.

Sun J., Kaban A. and Raychaudhury S. (2007). Robust visual mining of data with error information. *PKDD 2007* (Warsaw, Poland), SLNCS 4702, 573-580.

Valdes J. J. and Barton A. J. (2007). Finding relevant attributes in high dimensional data: A distributed computing hybrid data mining strategy. *Transactions on Rough Sets VI*, SLNCS 4374, 366-396.

Valdes J. J., Romero E. and Gonzalez R. (2007). Data and knowledge visualization with virtual reality spaces. *IJCNN 2007* (Orlando, USA), 160-165.

Zelenyuk, A., et al. (2006). SpectraMiner, an interactive data mining and visualization software for single particle mass spectroscopy: A laboratory test case, *International Journal of Mass Spectrometry*, *258*, 58-73.

KEY TERMS

Collaborative Virtual Environment: An environment that actively supports human-human communication in addition to human-machine communication and which uses a virtual environment as the user interface.

Curse of Dimensionality: The problems associated with information overload, when the number of dimensions is too high to visualize.

Data Visualization: The visualization of the data set through the use of a techniques such as scatter plots, 3D cubes, link graphs and surface charts.

Glyph: A three-dimensional object suitable for representing data or subsets of data.

Multimedia Data Mining: The application of data mining to data sets consisting of multimedia data, such as 2D images, 3D objects, video and audio. Multimedia data can be viewed as integral data records, which consist of relational data together with diverse multimedia content.

Visualization: The graphical expression of data or information.

Visual Data Mining: The integration of data visualization and data mining. Visual data mining is closely related to computer graphics, multimedia systems, human computer interfaces, pattern recognition and high performance computing.

Visualization of High-Dimensional Data with Polar Coordinates

Frank Rehm
German Aerospace Center, Germany

Frank Klawonn
University of Applied Sciences Braunschweig/Wolfenbuettel, Germany

Rudolf Kruse
University of Magdenburg, Germany

INTRODUCTION

Many applications in science and business such as signal analysis or costumer segmentation deal with large amounts of data which are usually high dimensional in the feature space. As a part of preprocessing and exploratory data analysis, visualization of the data helps to decide which kind of data mining method probably leads to good results or whether outliers or noisy data need to be treated before (Barnett & Lewis, 1994; Hawkins, 1980). Since the visual assessment of a feature space that has more than three dimensions is not possible, it becomes necessary to find an appropriate visualization scheme for such data sets.

Multidimensional scaling (MDS) is a family of methods that seek to present the important structure of the data in a reduced number of dimensions. Due to the approach of distance preservation that is followed by conventional MDS techniques, resource requirements regarding memory space and computation time are fairly high and prevent their application to large data sets. In this work we will present two methods that visualize high-dimensional data on the plane using a new approach. An algorithm will be presented that allows applying our method on larger data sets. We will also present some results on a benchmark data set.

BACKGROUND

Multidimensional scaling provides low-dimensional visualization of high-dimensional feature vectors (Kruskal & Wish, 1978; Borg & Groenen, 1997). MDS is a method that estimates the coordinates of a set of objects in a feature space of specified (low) dimensionality that come from data trying to preserve the distances between pairs of objects. In the recent years much research has been done (Chalmers, 1996; Faloutsos & Lin, 1995; Morrison, Ross, & Chalmers, 2003; Williams & Munzner, 2004; Naud, 2006). Different ways of computing distances and various functions relating the distances to the actual data are commonly used. These distances are usually stored in a distance matrix. The estimation of the coordinates will be carried out under the constraint, that the error between the distance matrix of the data set and the distance matrix of the corresponding transformed data set will be minimized. Thus, different error measures to be minimized were proposed, i.e. the absolute error, the relative error or a combination of both. A commonly used error measure is the so-called Sammon's mapping (Sammon, 1969). To determine the transformed data set by means of minimizing the error a gradient descent method is used.

Many modifications of MDS are published so far, but high computational costs prevent their application to large data sets (Tenenbaum, de Silva, & Langford, 2000). Besides the quadratic need of memory, MDS, as described above is solved by an iterative method, expensive with respect to computation time, which is quadratic in the size of the data set. Furthermore, a completely new solution must be calculated, if a new object is added to the data set.

MAIN FOCUS

With MDS_{polar} and POLARMAP we present two approaches to find a two-dimensional projection of a p-dimensional data set X. Both methods try to find a rep-

resentation in polar coordinates $Y = \{(l_1, \phi_1), \ldots, (l_n, \phi_n)\}$, where the length l_k of the original vector x_k is preserved and only the angle φ_k has to be optimized. Thus, our solution is defined to be optimal if all angles between pairs of data objects in the projected data set Y coincide as good as possible with the angles in the original feature space *X*. As we will show later, it is possible to transform new data objects without extra costs.

MDS$_{\text{polar}}$

A straight forward definition of an objective function to be minimized for this problem would be,

$$E = \sum_{k=2}^{n} \sum_{i=1}^{k-1} \left(\left| \phi_i - \phi_k \right| - \psi_{ik} \right)^2 \quad (1)$$

where φ_k is the angle of y_k, ψ_{ik} is the positive angle between x_i and x_k. The absolute value is chosen in equation (1) because the order of the minuends can have an influence on the sign of the resulting angle. The problem with this notation is that the functional *E* is not differentiable, exactly in those points we are interested in, namely, where the difference between angles φ_i and φ_k becomes zero.

We propose an efficient method that enables us to compute an approximate solution for a minimum of the objective function (1) and related ones. In a first step we ignore the absolute value in (1) and consider

$$E = \sum_{k=2}^{n} \sum_{i=1}^{k-1} \left(\phi_i - \phi_k - \psi_{ik} \right)^2. \quad (2)$$

When we simply minimize (2), the results will not be acceptable. Although the angle between y_i and y_k might perfectly match the angle ψ_{ik}, $\varphi_i - \varphi_k$ can either be ψ_{ik} or $-\psi_{ik}$. Since we assume that $0 \le \psi_{ik}$ holds, we always have $\left(\left| \phi_i - \phi_k \right| - \psi_{ik} \right)^2 \le \left(\phi_i - \phi_k - \psi_{ik} \right)^2$. Therefore, finding a minimum of (2) means that this is an upper bound for the minimum of (1). Therefore, when we minimize (2) in order to actually minimize (1), we can take the freedom to choose whether we want the term $\phi_i - \phi_k$ or the term $\phi_k - \phi_i$ to appear in (2). Since

$$\begin{aligned} &\left(\phi_i - \phi_k - \psi_{ik} \right)^2 \\ &= \left(-\left(\phi_i - \phi_k - \psi_{ik} \right) \right)^2 \\ &= \left(\phi_k - \phi_i + \psi_{ik} \right)^2 \end{aligned}$$

instead of exchanging the order of φ_i and φ_k, we can choose the sign of ψ_{ik}, leading to

$$E = \sum_{k=2}^{n} \sum_{i=1}^{k-1} \left(\phi_i - \phi_k - a_{ik} \psi_{ik} \right)^2 \quad (3)$$

with $a_{ik} \in \{-1, 1\}$.

In order to solve this optimization problem of equation (3) we take the partial derivatives of *E*, yielding

$$\frac{\partial E}{\phi_k} = -2 \sum_{i=1}^{k-1} \left(\phi_i - \phi_k - a_{ik} \psi_{ik} \right). \quad (4)$$

Thus, on the one hand, neglecting that we still have to choose a_{ik}, our solution is described by a system of linear equations which means its solution can be calculated directly without the need of any iteration procedure. On the other hand, as described above, we have to handle the problem of determining the sign of the ψ_{ik} in the form of the a_{ik}-values. To fulfill the necessary condition for a minimum we set equation (4) equal to zero and solve for the φ_k-values, which leads to

$$\phi_k = \frac{\sum_{i=1}^{k-1} \left(\phi_i - a_{ik} \psi_{ik} \right)}{k-1}. \quad (5)$$

Since we only want to preserve the angles between data vectors, it is obvious that any solution will be invariant with respect to rotation of the data set. Due to the representation in polar coordinates it is necessary to apply a preprocessing step in form of a translation that makes all components of data vectors non-negative. Reasons for that and further details are given in (Rehm, Klawonn, & Kruse, 2005).

A Greedy Algorithm for the Approximation of MDS$_{\text{polar}}$

As mentioned above, this solution describes a system of linear equations. Since the desired transformation is rotation invariant φ_1 can be set to any value, i.e. $\varphi_1 = 0$. By means of a greedy algorithm we choose $a_{ik} \in \{-1, 1\}$ such that for the resulting φ_k the error *E* of the objective function (3) is minimal. For φ_2 the exact solution can always be found, since a_{12} is the only parameter to optimize. For the remaining φ_k the greedy algorithm sets a_{ik} in turn either –1 or 1, verifying the validity of the result, setting a_{ik} the better value immediately and continuing with the next a_{ik} until all k–1 values for a_{ik} are set.

A Generalized MDS_{polar}

In certain cases the objective when transforming data is to preserve relations of feature vectors of the original feature space in the target feature space. Thus, feature vectors that form a cluster should be represented as exact as possible in the target feature space, too. The transformation of feature vectors with a large distance to the respective feature vector can have a lower accuracy. An approach to achieve this goal is the introduction of weights w_{ik} to our objective function

$$E = \sum_{k=2}^{n} \sum_{i=1}^{k-1} w_{ik} \left(\phi_i - \phi_k - a_{ik} \ \psi_{ik} \right)^2. \quad (6)$$

The main benefit of weights, indeed, is the ability to decrease the computational complexity of the algorithm. This is the case if weights are chosen in such a way, that for feature vectors with a certain (large) distance the respecting weights become zero. A weighting function can control this behavior automatically. For an efficient implementation it is useful to sort the feature vectors by means of their length. Note that sorting can be carried out in less than quadratic time. Weighting functions should be decreasing and should lead to zero weights for proper feature vectors. Different weighting functions and further details can be seen in (Rehm, Klawonn, & Kruse, 2005). In this way, feature vectors can be grouped into suitable bins, reducing the complexity of our algorithm to $O(n \cdot \log n)$.

POLARMAP

As an extension of MDS_{polar} we propose in this work a method that learns a function f that provides for any p-dimensional feature vector x_k the corresponding angle φ_k that is needed to map the feature vector to a 2-dimensional feature space. As for MDS_{polar} the length of vector x_k is preserved. With the obtained function also angles for new feature vectors can be computed. A 2-dimensional scatter plot might not be suitable, when visualizing mappings for large data sets. With the computed function it is simple to produce information murals, which allow more comprehensive visualizations (Jerding & Stasko, 1995).

Analogous to functional (1) we define our objective function E as follows:

$$E = \sum_{k=2}^{n} \sum_{i=1}^{k-1} \left(\left| f(x_i) - f(x_k) \right| - \psi_{ik} \right)^2. \quad (6)$$

Since functional (6) is not differentiable, we propose analogous to the procedure for MDS_{polar} to minimize the following differentiable objective function

$$E = \sum_{k=2}^{n} \sum_{i=1}^{k-1} \left(f(x_i) - f(x_k) - \psi_{ik} \right)^2. \quad (7)$$

Albeit, f might be any function, we discuss in this work the following type of function

$$f(x) = a^T \cdot \tilde{x} \quad (8)$$

where a is vector whose components are the parameters to be optimized and $\tilde{x}$ is the feature vector x itself or a modification of x. In the simplest case we use

$$\begin{aligned} \tilde{x} &= x \\ a &= (a_1, a_2, \ldots, a_p) \end{aligned} \quad (9)$$

where f describes in fact the linear combination of x. Other functions f are discussed in (Rehm, Klawonn, & Kruse, 2006).

Replacing term f by the respective function we obtain

$$E = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \left(a^T \tilde{x}_i - a^T \tilde{x}_j - \psi_{ik} \right)^2$$

$$= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \left(a^T (\tilde{x}_i - \tilde{x}_j) - \psi_{ik} \right)^2. \quad (10)$$

For a better readability we replace $\tilde{x}_i - \tilde{x}_j$ by $\tilde{x}_{ij}$ and obtain

$$E = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \left(a^T \tilde{x}_{ij} - \psi_{ij} \right)^2. \quad (11)$$

The derivative of E w.r.t. a can by easily obtained

$$E = 2 \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \left(a^T \tilde{x}_{ij} - \psi_{ij} \right) \tilde{x}_{ij} \quad (12)$$

which results in a system of linear equations in $a = \left(a_1, a_2, \ldots, a_p \right)^T$. As mentioned already, angles computed by $f(x_i) - f(x_j)$, might be positive or negative, while ψ_{ik} is always positive by definition. Thus, in the case where $a^T \tilde{x}_{ij} < 0$ holds, E might be minimal, but our original objective function E might not be minimal.

Figure 1. Sammon's Mapping of the Wisconsin Breast Cancer Dataset

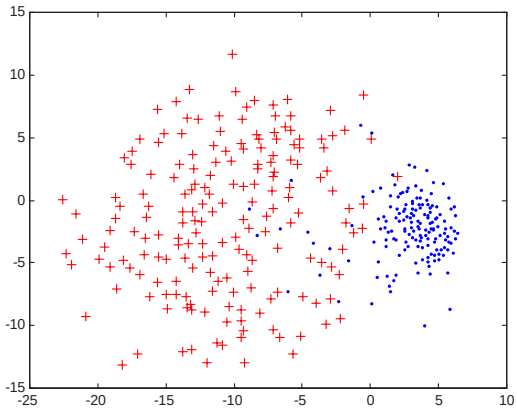
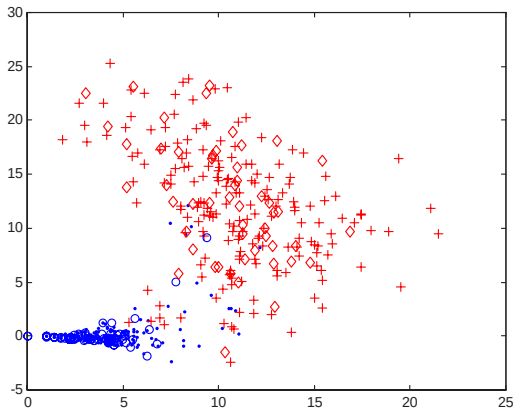


Figure 2. POLARMAP on the Wisconsin Breast Cancer Dataset



Hence, replacing $\tilde{x}_{ij}$ by $-\tilde{x}_{ij}$ in this case might lower the error. Consequently, finding the appropriate sign for $\tilde{x}_{ij}$ is a crucial step when minimizing E.

Determining the sign for each $\tilde{x}_{ij}$ requires exponential need of computation time in the number of feature vectors. For real-world data sets this is unacceptable. When relaxing the problem in favor to an approximation of the exact solution one can reduce the time complexity down to $O(n \cdot \log n)$. As for $\text{MDS}_{\text{polar}}$ this can be achieved by means of introducing weights which is discussed in detail in (Rehm, Klawonn, & Kruse, 2006).

Experimental Results

Since a function is learned by POLARMAP it becomes possible to map new vectors in the target space. To demonstrate the power of POLARMAP, we applied it on a UCI database, the well known Wisconsin breast cancer data set[1] (Mangasarian & Wolberg, 1990). Each patient in the database had a fine needle aspirate taken from her breast. Resultant, nine attributes where determined and analyzed to discriminate benign from malignant breast lumps. The original number of instances is about 699 that can be reduced to 481 data by removing redundant measurements or measurements that have missing values. Each instance in the data set is described by means of ten numerical attributes and one additional class attribute that classifies the measurement either benign or malignant accordingly.

Figure 1 shows the Sammon's mapping of the data set. The transformation of the Wisconsin breast cancer data set with POLARMAP is shown in figure 2. The different classes are represented by different symbols. Whereas the plus symbol corresponds to a benign

sample, does the dot symbol represent a malignant sample. Both transformations are similar regarding the scattering of the different classes. Patients with benign lumps and those with malignant lumps can be almost separated linearly in both transformations. Only few points can be found in regions where the opposite class mainly represented. Note, labeling of the axes is omitted intentionally in the figures. Since the discussed visualization techniques arrange some kind of non-trivial dimension reduction both axes represent the original attributes in a composite manner.

For the transformation with POLARMAP, the data set is split into a training data set and a test data set. The training data set consists of 80% of each class. This part of the data is used to learn the desired coefficients. The test data set, that contains the remaining 20% of the data, is mapped to the target space by means of the learned function. The mapping of the training data set is plotted with the different symbols again, each for the corresponding class. The mapped feature vectors of the test data set are marked with a small circle (for malignant samples) or a diamond (for benign samples), respectively. As the figure shows, the learned function maps the new feature vectors in an appropriate way.

FUTURE TRENDS

We presented two approaches that provide visualization of high-dimensional data. MDS_{polar} and POLARMAP find 2-dimensional representations of complex data trying to preserving angles between feature vectors. The bin-strategy that uses a weighting function to reduce the complexity of the algorithm, benefits from sorting the feature vectors by means of their length. Indeed, sorting guarantees that short feature vectors are quite similar. Nevertheless, longer feature vectors can be either similar or different, namely if they have different directions. Thus, another sorting criterion could improve the results of both proposed visualization methods. Appropriate techniques should be subject of future work.

CONCLUSION

In this paper we have described a powerful data visualization method. Under the constraint to preserve the length of feature vectors, it was our aim to find a mapping that projects feature vectors from a high-dimensional space to the plane in such a way that we minimize the errors in the angles between the mapped feature vectors. The solution is described by a system of linear equations. To overcome the problem in high-dimensional feature spaces, that no differentiation between positive and negative angles can be made as for a 2-dimensional feature space, an algorithm is provided to obtain the desired signs for the angles. With the bin-algorithm we presented an algorithm that lowers the computation complexity down to $O(n \cdot \log n)$. Experimental results on a benchmark data set have shown that the results of the proposed techniques are comparable to those of conventional MDS but fewer resources will be needed. In particular when additional instances have to be mapped, the proposed techniques can achieve this without extra costs.

REFERENCES

Barnett, V., & Lewis, T. (1994). *Outliers in statistical data*. John Wiley & Sons, New York.

Borg, I., Groenen, P. (1997). *Modern Multidimensional Scaling: Theory and Applications.* Springer, Berlin.

Chalmers, M. (1996). *A Linear Iteration Time Layout Algorithm for Visualising High-Dimensional Data.* Proceedings of IEEE Visualization, San Francisco, CA, 127-132.

Faloutsos, C., & Lin, K. (1995). *Fastmap: A Fast Algorithm for Indexing, Data-Mining and Visualization of Traditional and Multimedia Datasets.* Proceedings of ACM SIGMOD International Conference on Management of Data, San Jose, CA, 163-174.

Hawkins, D. (1980). *Identification of outliers*. Chapman & Hall, London.

Kruskal, J. B., & Wish, M. (1978). *Multidimensional Scaling*. SAGE Publications, Beverly Hills.

Jerding, D. F., & Stasko, J. T. (1995). *The information mural: a technique for displaying and navigating large information spaces.* Proceedings of the 1995 IEEE Symposium on Information Visualization, 43–50.

Mangasarian, O.L., & Wolberg, W.H. (1990). Cancer diagnosis via linear programming. *SIAM News, 23*(5), 17-18.

Morrison, A., Ross, G., & Chalmers, M. (2003). Fast Multidimensional Scaling through Sampling, Springs and Interpolation. *Information Visualization, 2,* 68-77.

Naud, A. (2006). An accurate MDS-based algorithm for the visualization of large multidimensional datasets. In Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A. & Zurada, J. (Editors), *Proceedings of the 8th International Conference on Artificial Intelligence and Soft Computing (ICAISC 2006) (pp. 643–652)*. Lecture Notes in Computer Science 4029, Springer.

Rehm, F., Klawonn, F., & Kruse, R. (2005). MDS_{polar} - A New Approach for Dimension Reduction to Visualize High Dimensional Data. In: Advances in Intelligent Data Analysis VI: 6th International Symposium on Intelligent Data Analysis, IDA 2005, Springer, 316-327.

Rehm, F., Klawonn, F., & Kruse, R. (2006). POLARMAP—Efficient Visualization of High Dimensional Data. *IEEE Proceedings of the 10th International Conference on Information Visualisation,* London, 2006.

Sammon, J.W. (1969). A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers, 18,* 401-409.

Tenenbaum, J. B., de Silva, V., & Langford, J. C. (2000). A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science, 290,* 2319–2323.

UCI repository of machine learning databases and domain theories. www.ics.uci.edu/~mlearn.

Williams, M., & Munzner, T. (2004). Steerable, Progressive Multidimensional Scaling. 10th IEEE Symposium on Information Visualization, Austin, TX, 57-64.

KEY TERMS

Data Visualization: Presentation of data in human understandable graphics, images, or animation.

Information Mural: A technique that allows 2D visual representations of large information spaces to be created even when the number of informational elements greatly outnumbers the available resolution of the display device.

MDS_{polar}: A multidimensional scaling technique whose optimizations criterion is the minimization of the difference of angles between feature vectors.

Multidimensional Scaling: Provides low-dimensional visualization of high-dimensional feature vectors.

POLARMAP: A modification of MDS_{polar} in that point that a function is learned to map high-dimensional feature vectors on the plane. By means of this function even new feature vectors can be mapped without any extra costs.

Sammon's Mapping: An error measure of distances between feature vectors to be minimized for dimension reduction. A combination of both - the absolute error and the relative error - will be considered.

Visual Data Mining: Data mining process through data visualization. The fundamental concept of visual data mining is the interaction between data visual presentation, human graphics cognition, and problem solving.

ENDNOTE

[1] The breast cancer database was obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg.

Visualization Techniques for Confidence Based Data

Andrew Hamilton-Wright
University of Guelph, Canada, & Mount Allison University, Canada

Daniel W. Stashuk
University of Waterloo, Canada

INTRODUCTION

Decision support algorithms form an important part of the larger world of data mining. The purpose of a decision support system is to provide a human user with the context surrounding a complex decision to be based on computational analysis of the data at hand. Typically, the data to be considered cannot be adequately managed by a human decision maker because of its volume, complexity or both; data mining techniques are therefore used to discover patterns in the data and inform the user of their saliency in terms of a particular decision to be made.

Visualization plays an important role in decision support, as it is through visualization that we can most easily comprehend complex data relationships (Tufte, 1997, 2001, 2006; Wright, 1997). Visualization provides a means of interfacing computationally discovered patterns with the strong pattern recognition system of the human brain. As designers of visualization for decision support systems, our task is to present computational data in ways that make intuitive sense based on our knowledge of the brain's aptitudes and visual processing preferences.

Confidence, in the context of a decision support system, is an estimate of the value a user should place in the suggestion made by the system. System reliability is the measure of overall accuracy; confidence is an estimate of the accuracy of the suggestion currently being presented. The idea of an associated confidence or certainty value in decision support systems has been incorporated in systems as early as MYCIN (Shortliffe, 1976; Buchanan & Shortliffe, 1984).

BACKGROUND

A decision support system functions by taking a set of rules and evaluating the most preferable course of action. The most preferable of a set of possible actions is chosen based on an internal optimization of some form of objective function. This optimization may take one of several forms: a full cost-benefit analysis (Rajabi, Kilgour & Hipel, 1998; Hipel & Ben-Haim, 1999); a simple best-rule match; or that of a multi-rule evaluation using rules weighted by their expected contribution to decision accuracy (Hamilton-Wright, Stashuk & Tizhoosh, 2007).

The underlying rules forming the structure of a decision support system may be found using an automated rule discovery system, allowing a measure of the quality of the pattern to be produced through the analysis generating the patterns themselves (Becker, 1968); in other cases (such as rules produced through interview with experts), a measure of the quality of the patterns must be made based on separate study (Rajabi, Kilgour & Hipel, 1998; Kononenko & Bratko, 1999; Kukar, 2003; Gurov, 2004a,b).

The construction of a tool that will assist in choosing a course of action for human concerns demands a study of the confidence that may be placed in the accurate evaluation of each possible course. Many of the suggestions made by a decision-support system will have a high-risk potential (Aven, 2003; Crouhy, Galai & Mark, 2003; Friend & Hickling, 2005). Examples of such systems include those intended for clinical use through diagnostic inference (Shortliffe, 1976; Buchanan & Shortliffe, 1984; Berner, 1988; de Graaf, van den Eijkel, Vullings & de Mol, 1997; Innocent, 2000; Coiera, 2003; Colombet, Dart, Leneveut, Zunino, Ménard & Chatellier, 2003; Montani, Magni, Bellazzi,

Larizza, Roudari & Carson, 2003; Devadoss, Pan & Singh, 2005) and medical informatics (Bennett, Casebeer, Kristofco & Collins, 2005): other systems may have a lower immediate risk factor, but the long term public risk may be extensive, such as in environmental planning and negotiation (Rajabi, Kilgour & Hipel, 1998; Freyfogle, 2003; Randolph, 2004).

In such high-risk cases, a user cannot proceed through a decision process with a blind trust in a suggested algorithmic solution. This observation is further supported by the consideration that the possible solutions promoted by the algorithm will have a broad variability in confidence support themselves: some courses of action will be suggested based on only the thinnest degree of support; others may have a large margin of error. The disparity between these cases makes it obvious that one would clearly be unwise to treat the two suggestions in the same way when incorporating the suggested algorithmic decision into a larger course of action. Ideally, suggestions associated with a lower degree of confidence will be ratified through some other form of external evidence before being put into action. Such corroboration is certainly more desirable in the case of the less-confident decision than that of the more-confident one. The use of confidence based metrics for decision quality analysis has been discussed in the context of decision support since the inception of the field (Morton, 1971; Sage, 1991; Silver, 1991; Hipel & Ben-Haim, 1999). In order to trigger this ratification, it must be clear to the user what the relative and specific confidence values associated with a suggestion are.

The intent of this article is to discuss methods for conveying confidence to a human decision maker, and introduce ideas for clearly presenting such information in the context of a larger discussion of system design and usability (Norman, 1998; Tufte, 2001, 2006). The discussion will be based on decision support within a computationally supported visualization context (Wright, 1997; Brath, 1997; 2003; Mena, 1999).

MAIN FOCUS

Given that confidence in decision-making is a necessary and central concept to communicate to a human user, it is of interest to study how this concept may be conveyed. Counter-intuitively, although the concept of confidence is a central concept of decision support, an unambiguous formalism of confidence is lacking, due to the fact that different representations may or may not take into account a potential two-class labeling outcome in the confidence representation.

Confidence as Probability

In this representation, confidence is simply the perceived probability of a correct suggestion. The range of such a confidence measure is therefore [0…1], with the implication being that for a two-outcome case, a value of 0.5 will indicate "even odds", or a 50%-confidence solution. The value of this point will vary depending on the number of outcomes possible in the decision system at hand, and therefore the choice of this mechanism of confidence representation must be weighed against the system clarity of the number of outcomes for a decision will vary, especially if the number of outcomes may be substantially large.

The strongest reason to choose this representation of confidence is the direct relationship to probability; this clear relationship will aid decision makers with a strong statistical background, and will enable the confidence value to be integrated into a larger decision with greater ease and reliability.

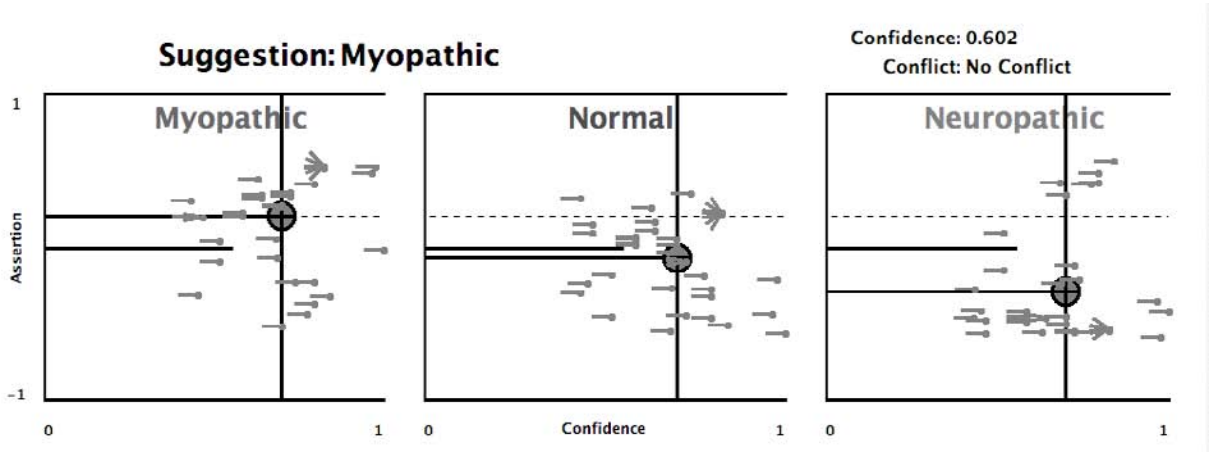
A further strength of this representational choice is the ability to have different probabilistic confidence values associated with different outcomes; these can be transparently calculated and clearly represented as part of the summary visualization for each case (see example below).

Confidence as Distance

This representation holds that confidence is a quality that has only a positive measure; that is the degree of confidence a system has in a given outcome is measured on a [0…1] scale where 0 indicates random chance and 1 indicates perfect certainty.

Figure 1 shows a sample visualization taken from a decision support tool (Hamilton-Wright & Stashuk, 2006; Hamilton-Wright, Stashuk & Tizhoosh, 2007). This tool produces "assertions" based on a rule weighting, and calculates a confidence value for each assertion produced. Suggested outcomes are proposed as aggregates of assertion weighting and confidence value. The location of the ticks shown in Figure 1 is therefore

Figure 1. Confidence visualization



important, as each of the three co-ordinate spaces relates aggregate decision support to both assertion weighting and confidence.

The figure shown here demonstrates the inclusion of all of these elements:

- each underlying data point is shown with a small tick; the *y* location of the tick indicates a degree of assertion associated with each point, and the *x* location indicates a confidence value in the [0…1] bounded range.
- an aggregate assertion is shown at a *y* location generated by an overall assertion value; these are shown by the larger circular marks.
- the overall confidence value is shown by the *x* location of *each* aggregate value.

By incorporating confidence and assertion weighting together, one may see at a glance to what degree the logic of the assertion weights are reinforced by their associated confidence values. It is clear in Figure 1 that the only high-confidence points are those whose assertion is significantly far from the zero line.

This display strategy brings up several points of note:

- the confidence display, being shown as a point along the *x*-axis of a bounded grid, accurately reflects the bounded nature of the confidence data; further, the lack of a central reference to the [0…1] number line reflects (accurately) that confidence in this sense is represented as a "degree of merit" associated with the indicated datum.
- the choice of *x* location reflects the continuous nature of the confidence data, as it is implied that all values between 0 and 1 are legal locations; the use of a linear color gradient would additionally support such a meaning.
- by using a one-dimensional representation for confidence, we are free to combine confidence values with another displacement-based data value – in this case, an assertion value was chosen. This allows the user to examine the relationship between confidence and assertion; as can be seen in the display, there is a general trend along the relation $x = |y|$ in each graph, which provides a useful insight by allowing a user to see a (desired) data dependency between confidence and assertion support.
- the strongest negative statement to be made about this display is the re-representation of confidence in each outcome domain; the display uses the same *x* location for all three confidence values. This is done as there is only a single confidence value available, however the obvious implication is the opposite: that there are three different confidence

values that are co-incidentally the same. Whether the strength of co-plotting confidence overcomes this undesired consequence will depend on the domain in question.

FUTURE TRENDS

Decision confidence is a necessary condition for intelligent decision support systems. The need for integration of confidence values into visual landscapes remains central to the requirements for decision support. Future visualization paradigms and tools will explore new ways to consistently and transparently integrate confidence visualization into the context of the data landscape supporting the decision to be made.

By moving to three-dimensional visualization (Wright, 1997; Mena, 1999) further degrees of freedom are introduced; providing further opportunities for the plotting of confidence values as supporting but independent data values. Obvious extensions of the above-mentioned ideas are to use the depth of a data bar for confidence while length and color may represent data values. Care must be taken to avoid overloading the decision maker with too much information, especially through displays containing redundant or extraneous information; Tufte (2001) urges designers to avoid "non-data" or "redundant data ink", and a consistent theme throughout his works asks for thought on the representation of data through clear use of strategically chosen representations (Tufte 1997, 2001, 2007).

Further papers regarding the representation of confidence and probabilistic error are being produced (Rossi, 2006; Sun, Kabán & Raychaudhury, 2007), as this study is central to the development of robust and understandable visual decision support systems.

CONCLUSIONS

Confidence values must accompany suggested courses of action in any decision support system. By achieving a coupled representation of confidence and suggestion, a system may provide insightful visualization that is easily understood by a decision-making user. Current visualization techniques provide a number of interesting tools to explore various methods of achieving this goal; one of the strongest remains the use of distance and position.

REFERENCES

Aven, T. (2003). *Foundations of Risk Analysis: A Knowledge and Decision Oriented Perspective.* Wiley.

Becker, P.W. (1968). *Recognition of Patterns: Using the Frequencies of Occurrence of Binary Words*. 2nd edition. Springer.

Bennett, N.L., Casebeer, L.L., Kristofco, R. & Collins, B.C. (2005). Family physicians' information seeking behaviors: A survey comparison with other specialities. *BMC Medical Informatics and Decision Making*, 5(9).

Berner, E.S., (Ed.). (1988). *Clinical Decision Support Systems: Theory and Practice*. Springer.

Brath, R. (1997, June). *3D interactive information visualization: Guidelines from experience and analysis of applications*. In 4th International Conference on Human--Computer Interaction.

Brath, R. (2003). Paper landscapes: A visualization design methodology. In Erbacher, R.F., Chen, P.C., Roberts, J.C. , Groehn M.T. & Boerner, K. editors. *Visualization and Data Analysis.* 5009. International Society for Optical Engineering (SPIE).

Buchanan, B.G. & Shortliffe, E. H. (Ed.). (1984). *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project.* Addison-Wesley.

Coiera, E. (2003). Guide to Health Informatics. 2nd edition. London: Arnold/Hodder & Stoughton.

Colombet, I., Dart, T., Leneveut, L. Leneveut, S., Ménard, J. & Chatellier, G. (2003). A computer decision aid for medical prevention: A pilot qualitative study of the personalized estimate of risks (EsPeR) system. *BMC Medical Informatics and Decision Making*, 3(13).

Crouhy, M., Galai, D. & Mark, R. (2003). *Risk Management.* McGraw-Hill.

de Graaf, P. M. A., van den Eijkel, G. C., Vullings, H.J.L.M. & de Mol, B.A.J.M. (1997). A decision-driven design of a decision support system in anaesthesia. *Artificial Intelligence In Medicine*. 11, 141-153.

Devadoss, P. R., Pan, S. L. & Singh, S. (2005). Managing knowledge integration in a national health-care crisis: Lessons learned from combating SARS in Singapore.

IEEE Transactions on Information Technology in Biomedicine. 9(2), 266--275.

Friend, J. & Hickling, A. (2005). *Planning Under Pressure.* 3rd edition. Elsevier.

Freyfogle, E.T. (2003). *The Land We Share: Private Property and the Common Good.* Washington/Covelo/London: Island Press/Shearwater Books.

Gurov, S. I. (2004a). Reliability estimation of classification algorithms I. *Computation Mathematics and Modeling*, 15(4), 365--376.

Gurov, S. I. (2004b). Reliability estimation of classification algorithms II. *Computation Mathematics and Modeling*, 16(2):169--178.

Hipel, K. W. & Ben-Haim, Y. (1999). Decision making in an uncertain world: Information-gap modeling in water resources management. *IEEE Transactions Systems, Man, Cybernetics C*. 29(4), 506--517.

Innocent, P. R. (July 2000). Fuzzy symptoms and a decision support index for the early diagnosis of confusable diseases. In *Proceedings of the RASC Conference*. Dept. of Computing Science, De Montfort University, Leicester, UK. 1-8.

Kononenko, I. & Bratko, I. (1991). Information-based evaluation criterion for classifier's performance. *Machine Learning*. 6, 67-80.

Kukar, M. (2003). Transductive reliability estimation for medical diagnosis. *Artificial Intelligence In Medicine*, 29, 81-106.

Mena, J. (1999). *Data Mining Your Website.* Boston: Digital Press.

Montani, S., Magni, P., Bellazzi, R. Larizza, C., Roudari, A.V. & Carson, E.R. (2003). Integrating model-based decision support in a multi-modal reasoning system for managing type 1 diabetic patients. *Artificial Intelligence In Medicine*, 29, 131-151.

Morton, S. (1971). *Management Decision Systems: Computer-Based Support for Decision Making*. Harvard University Press.

Norman, D.A. (1998). *The Design of Everyday Things.* Basic Books. (Former Title: The Psychology of Everyday Things).

Rajabi, S., Kilgour, D.M. & Hipel, K. W. (1998). Modeling action-interdependence in multiple criteria decision making. *European Journal of Operations Research*, 110(3), 490-508.

Randolph, J. (2004). *Environmental Land Use Planning and Management.* Washington/Covelo/London: Island Press/Shearwater Books.

Rossi, F. (2006). Visual Data Mining and Machine Learning. In *Proceedings of XIVth European Symposium on Artificial Neural Networks (ESANN 2006)*, 251-264, Bruges, Belgium.

Sage, A.P. (1991) *Decision Support Systems Engineering*. Wiley Series in Systems Engineering. John Wiley & Sons.

Shortliffe, E.H. (1976). *Computer-Based Medical Consultations: MYCIN.* Elsevier/North-Holland.

Silver, M.S. (1991). *Systems That Support Decision Makers: Description and Analysis*. John Wiley Information Systems Series. John Wiley & Sons.

Sun, J. Kabán A., & Raychaudhury, S. (2007). Robust visual mining of data with error information. I *11th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD07)*, Warsaw, Poland.

Tufte, E.R. (1997). *Visual Explanations: Images and Quantities, Evidence and Narrative*. Graphics Press, Cheshire, Connecticut.

Tufte, E.R. (2001). *The Visual Display of Quantitative Information*. 2nd edition. Graphics Press, Cheshire, Connecticut.

Tufte, E.R. (2006). *Beautiful Evidence*. Graphics Press, Cheshire, Connecticut.

Wright, W. (1997). Multi-dimensional representations -- how many dimensions? In *New Paradigms in Information Visualization and Manipulation*. ACM.

KEY TERMS

Confidence Measure: A metric produced by a data analysis system representing the likelihood of error in an associated calculated value. In a decision support

system, this is frequently related to the probability that the suggested course of action is in fact correct or the best.

Data Cleaning: A data mining task in which input data is examined to ensure that erroneous values are unlikely to be included. In terms of confidence estimation, an estimate of how "clean" a data set is will play a part in determining the confidence placed in any rules generated by analysis of such data.

Data Visualization: The construction of a two- or three-dimensional scene that represents relationships within a data set using relationships based on concepts used by the human visual system. Such concepts include geometric placement, relative scale, color and orientation.

Decision Exploration: A task related to decision support, decision exploration refers to the ability of a data analysis system to provide feedback on the consequences of, and rationale behind, a particular course of action. By exploring several courses of action, a human user will become more aware of the relative merits of each course.

Decision Support: The process of mining and presenting background data for the purposes of assisting in a complex decision. Typically, this is done by computational means to speed the process of data analysis, but may also be done through human collaboration. In all cases, the objective is to make clear to a decision maker the evidence in support of, or against, any particular course of action.

High-Risk Decision Making: An activity in which it is of greater than normal importance to ensure that poor outcomes are not chosen. In a high-risk environment, it is significantly preferable to determine the outcome of potential decisions by decision exploration using a model rather than direct experimentation. Many applications in civil engineering, public health and environmental management are considered to be "high risk" areas requiring special tools to avoid detrimental courses of action.

Risk Assessment: A protocol that will generate an estimate of the risk associated with a given course of action. This differs from confidence in that confidence is an estimate of the probability of an error being made; a risk is a probability that the outcome may be unfavorable in the case where all estimates are correct. A system may estimate a 7 in 10 chance of success with a 95% confidence interval; the risk in this case would be 3 in 10.

Visual Representation of Statistical Information: A mapping provided from a probabilistic data element into a visual representation. Statistical information is defined in terms of a (bounded) portion of a real-number line, and therefore a continuous-valued representation is preferred: typically this is done with a location-based metaphor with an associated axis, or color. Representations that imply enumeration, counts or other discrete phenomena are to be avoided.

Web Design Based on User Browsing Patterns

Yinghui Yang
University of California, Davis, USA

INTRODUCTION

It is hard to organize a website such that pages are located where users expect to find them. Consider a visitor to an e-Commerce website in searching for a scanner. There are two ways he could find information he is looking for. One is to use the search function provided by the website. The other one is to follow the links on the website. This chapter focuses on the second case. Will he click on the link "Electronics" or "Computers" to find the scanner? For the website designer, should the scanner page be put under Electronics, Computers or both? This problem occurs across all kinds of websites, including B2C shops, B2B marketplaces, corporate web-sites and content websites.

Through web usages mining, we can automatically discover pages in a website whose location is different from where users expect to find them. This problem of matching website organization with user expectations is pervasive across most websites. Since web users are heterogeneous, the question is essentially how to design a website so that majority of the users find it easy to navigate. Here, we focus on the problem of browsing within a single domain/web site (search engines are not involved since it's a totally different way of finding information on a web site.) There are numerous reasons why users fail to find the information they are looking for when browse on a web site. Here in this chapter, we focus on the following reason. Users follow links when browsing online. Information scent guides them to select certain links to follow in search for information. If the content is not located where the users expect it to be, the users will fail to find it. How we analyze web navigation data to identify such user browsing patterns and use them to improve web design is an important task.

BACKGROUND

There has been considerable amount of work on mining web logs – Web Usage Mining. Web usage mining is a viable framework for extracting useful access pattern information from massive amounts of web log data for the purpose of web site personalization and organization (Missaoui et al 2007; Srivastava 2000; Nasraoui et al. 2003; Mobasher & Anand 2005). Various tasks can be achieved via web usage mining (e.g., finding frequent and interesting navigation patterns, predicting future page requests and page recommendations). Spiliopoulou et al. (1998) and Spiliopoulou et al. (1999) propose a "web utilization miner" (WUM) to find interesting navigation patterns. The interestingness criteria for navigation patterns are dynamically specified by the human expert using WUM's mining language. Chen & Cook (2007) proposes a new data structure for mining contiguous sequential patterns from web log. Liu et al (2007) presents a study of the automatic classification of web user navigation patterns and propose an approach to classifying user navigation patterns and predicting users' future requests. The approach is based on the combined mining of web server logs and the contents of the retrieved web pages.

A small subset of the research in web usage mining uses the usage patterns observed from the web logs to improve web design. There is quite some research on link recommendations based on users' previous browsing patterns, mainly utilizing the sequence of pages visited. Perkowitz et al. (1998) and Perkowitz et al. (1999) investigate the problem of index page synthesis, which is the automatic creation of pages that facilitate a user's navigation of a website. By analyzing the web log, their cluster mining algorithm finds collections of pages that tend to co-occur in visits and puts them under one topic. They then generate index pages consisting of links to pages pertaining to a particular topic. Baraglia & Silvestri (2007) introduces a web personalization system that is online and incremental and it is designed to dynamically generate personalized contents of potential interest for users of large web sites made up of pages dynamically generated. It is based on an incremental personalization procedure tightly coupled with the web server. It is able to update incrementally and automatically the knowledge base

based on the ontology underlying the site. Data mining techniques can be applied to enriched web logs to extract knowledge that could be used to improve the navigational structure as well as exploited in recommender systems.

Another stream of related research is on Information foraging theory (Pirolli & Card 1995; Pirolli & Fu 2003). As mentioned in Juvina & Herder (2005), web sites generally are designed for a general audience with varying goals. As it is hard to satisfy all categories of users with one design, adaptive hypermedia systems try to better support the users by personalizing content or link structure. Traditional techniques in the latter category involve link hiding, sorting, annotation, direct guidance and hypertext map adaptation (Brusilovsky 2001). When trying to find information related to a task, users have to rely on proximal cues such as the link anchor text to decide what their next action will be. If the proximal cues are not clear enough, or if the users do not have sufficient insight on the structure of the site, they may become disoriented, i.e. they don't know their current position in a web site, how they came to that point or where to go next (Herder & Juvina 2004). Various studies have been carried out to infer user goals from their actions (e.g. Chi et al. 2003). Given these goals, the utility of the various navigation options on a web page can be estimated (Kitajima et al. 2000; Pirolli & Fu 2003) and communicated to the user by means of link relevancy indicators, or link suggestions.

Also, research on users' page revisit behavior has documented that users frequently visit pages already visited before. Earlier studies (Catledge & Pitkow 1995; Tauscher & Greenberg 1997; Cockburn & McKenzie 2001) have shown that the majority of page requests involve requests to pages visited before.) Tauscher & Greenberg (1997) identified the following main reasons for revisiting pages: (a) the information contained by them changes; (b) they wish to explore the page further; (c) the page has a special purpose; (d) they are authoring a page; (e) the page is on a path to another revisited page. When the revisited pages are home pages and index pages that serve to navigate users to a number of pages, the reason for backtracking can be also to search for information that's not found at the current location (Srikant & Yang 2001).

There is a great amount of research on locating documents via searching (Yeung et al 2007, Heflin et al 2003). Searching eventually becomes the dominating way of finding information when the web structure is too deep and complex to browse. Another stream of related research is automatic acquisition of taxonomies or concept hierarchies from a text corpus (Cimiano et al 2005). This can help organize documents automatically in a structure.

MAIN FOCUS OF THE CHAPTER

Our focus in this chapter is to address the gap between the web-site designer's expectations and the web users' expectations. The web designers' expectations are observed from the web site structure, and web users' expectations are observed from their usage patterns discovered for web logs. Through web usages mining, we can automatically discover users' traversal paths from which we can infer their expectation about how the web site should be structure.

We first briefly discuss several papers that address this problem and then focus more on one particular paper to illustrate a simple idea that can be used to improve the web design through users' behavioral patterns.

Nakayama et al. (2000) tries to discover the gap between the web-site designer's expectations and user behavior. Their approach uses the inter-page conceptual relevance to estimate the former, and the inter-page access co-occurrence to estimate the latter. They focus on website design improvement by using multiple regressions to predict hyperlink traversal frequency from page layout features. Paik et al (2002) describes the design and the implementation of a system through which existing on-line product catalogs can be integrated, and the resulting integrated catalogs can be continuously adapted and personalized within a dynamic environment. They propose a methodology for adaptation of integrated catalogs based on the observation of customers' interaction patterns. Gupta et al (2007) proposes a heuristic scheme based on simulated annealing that makes use of the aggregate user preference data to re-link the pages to improve navigability. Organizations maintain informational websites. The information content of such websites tends to change slowly with time, so a steady pattern of usage is soon established. User preferences, both at the individual and at the aggregate level, can then be gauged from user access log files. Their scheme is also applicable to the initial design of websites for wireless devices. Using the aggregate user preference data obtained from a parallel wired website, and given an upper bound on

the number of links per page, their methodology links the pages in the wireless website in a manner that is likely to enable the "typical" wireless user to navigate the site efficiently. Later, when a log file for the wireless website becomes available, the same approach can be used to refine the design further.

Srikant & Yang (2001) focuses on one type of web site structure, web sites with a top-down tree type of linkage structures, and looks at browsing activities within a web site. For this type of web sites, it is logical to consider the point from where the users backtrack the expected locations for the page they are looking for. Backtracking is the most common type of page revisitation and is both used for finding new information and relocating information visited before (Herder 2005). Under the assumption of backtracking browsing patterns, Srikant & Yang (2001) discusses the problem of automatically identifying pages in a website whose location is different from where users expect to find them and gives the algorithm for finding expected locations.

Search Strategies

When a user is looking for a single specific target page, we expect the user to execute the following search strategy:

1. Start from the root.
2. While (current location *C* is not the target page *T*) do
    a. If any of the links from *C* seem likely to lead to *T*, follow the link that appears most likely to lead to *T*.
    b. Else, either backtrack and go to the parent of *C* with some (unknown) probability, or give up with some probability.

The strategy the user takes when searching for set of targets is similar, except that after finding (or giving up on) one target, the user then starts looking for the next target.

Identify Target Pages

For some websites like Amazon and Ebay, there is a clear separation between content pages and index (or navigation) pages; product pages on these websites are content pages, and category pages are index or navigation pages. In such cases, we can consider the target pages for a visitor to be exactly the set of content pages requested by the visitor. Other websites such as information portals or corporate websites may not have a clear separation between content and index pages. For example, Yahoo! lists websites on the internal nodes of its hierarchy, not just on the leaf nodes. In this case, we can use a time threshold to distinguish whether or not a page is a target page. Pages where the visitor spent more time than the threshold are considered target pages. We can also combine these two methods, and have different time thresholds for different classes of pages.

Algorithm for Finding Expected Location

If there is no browser caching, it is conceptually trivial to find a backtrack point: it is simply the page where the previous and next pages in the web log (for this visitor) are the same. The HTTP standard states that the browser should not request the page again when using the browser's history mechanism. In practice, some browsers use the cached page when the visitor hits the "back" button, while others incorrectly request the page again. It is possible to disable caching by setting an expiration date (in the meta tag in the page), but this can significantly increase the load on the website. Rather than rely on disabling browser caching, we use the fact that if there is no link between pages P1 and P2, the visitor must have hit the "back" button in the browser to go from P1 to P2. Thus to find backtrack points, we need to check if there is a link between two successive pages in the web log. We build a hash table of the edges in the website to efficiently check if there is a link from one page to another.

In Step 1, we build the hash table. We partition the web log by visitor in Step 2. In Step 3, we split the sequence of accesses for each visitor by the target pages they visit. We assume that the website administrator either specifies the set of possible target pages, or specifies a time threshold to distinguish between target pages and other pages. In Step 4, we find all expected locations (if any) for that target page, and add it to a table for use by the next step of the algorithm. The detection of backtracks occurs in Step 4(b). In addition to checking for the absence of a link from the current to the next page, we also check if the previous and next pages are the same. The latter check takes

care of the case where visitors use a navigation link to go to the previous page instead of using the "back" button in the browser.

There are limitations to this approach. It can be hard to distinguish between target pages and other pages when the website does not have a clear separation between content and index pages. Hence the algorithm may generate false expected locations if it treats target pages as backtrack points, and may miss expected locations if it treats backtrack points as target pages. Increasing the time threshold will result in fewer missed expected locations at the cost of more false expected locations, while decreasing the threshold will have the opposite effect. Hence for websites without a clear separation between content and navigation, the administrator will have to spend some time to determine a good value for the time threshold, as well as sift through the discovered expected locations to drop any false patterns. Another limitation is that only people who successfully find a target page will generate an expected location for that page. We cannot track people who tried the expected location and gave up after not finding the target page.

Srikant & Yang (2001) also discussed several link recommendation strategies. (a) FirstOnly: Recommend all the pages whose frequency of occurrence in the first expected location is above an administrator-specified threshold. (b) OptimizeBenefit: Recommendthe set of pages that optimize benefit to the website, where benefit is estimated based on the fraction of people who might give up on not finding a page. (c) OptimizeTime: Recommend the set of pages that minimize the number of times the visitor has to backtrack, i.e., the number of times the visitor does not find the page in an expected location. The algorithm is applied to a university website, and we found many pages that were located differently from where users expected to fine them.

FUTURE TRENDS

An interesting problem for future research is that in websites without a clear separation of content and navigation, it can be hard to differentiate between users who backtrack because they are browsing a set of target pages, and users who backtrack because they are searching for a single target page. While Srikant & Yang (2001) has proposed using a time threshold to distinguish between the two activities, it will be interesting to explore if there are better approaches to solve this problem. Moreover, the web access patterns on a web site are very dynamic in nature, due not only to the dynamics of web site content and structure, but also to changes in the user's interests, and thus their navigation patterns. This is a huge challenge and needs to be carefully studied in the future.

CONCLUSION

In this chapter, research on using web usage mining to close the gap between the web designers' expectation and web users' expectation is surveyed. Through web usages mining, we can automatically discover pages in a website whose location is different from where users expect to find them. This problem of matching website organization with user expectations is pervasive across most websites.

REFERENCES

Baraglia, R. & Silvestri, F. (2007). Dynamic personalization of web sites without user intervention. Communications of the ACM, 50(2), pp. 63-67.

Brusilovsky, P. (2001). Adaptive Hypermedia. *User Modeling and User-Adapted Interaction 11*, 87-110.

Catledge, L.D. & Pitkow, J.E. (1995). Characterizing browsing strategies in the world wide web. *Computer Networks and ISDN Systems, 27(6)*, 1065–1073.

Chen, J. & Cook, T. (2007). Mining contiguous sequential patterns from web logs. In *Proceedings of the 16th international conference on World Wide Web*.

Chi, E.H., Rosien, A., Supattanasiri, G., Williams, A., Royer, C., Chow, C., Robles, E., Dalal, B., Chen, J. & Cousins, S. (2003). The Bloodhound Project: Automatic Discovery of Web Usability Issues using the InfoScent Simulator. In *Proceedings of CHI 2003*.

Cimiano, P. Hotho, A. & Staab, S. (2005). Learning Concept Hierarchies from Text Corpora using Formal Concept Analysis. *Journal of AI Research*, Volume 24: 305-339.

Cockburn, A. & McKenzie, B. (2001). What do web users do: An empirical analysis of web use. *Interna-*

tional Journal of Human-Computer Studies, 54(6), 903-922.

Gupta, R., Bagchi, A. & Sarkar, S. (2007). Improving Linkage of Web Pages. INFORMS Journal on Computing, 19(1), pp. 127-136.

Heflin, J., Hendler, J., & Luke, S. (2003). SHOE: A Blueprint for the Semantic Web. In Fensel, D., Hendler, J., Lieberman, H., and Wahlster, W. (Eds.), *Spinning the Semantic Web*. MIT Press, Cambridge, MA.

Herder, E. (2005). Characterizations of User Web Revisit Behavior. In *Workshop on Adaptivity and User Modeling in Interactive Systems*, 32-37.

Herder, E. & Juvina, I. (2004). Discovery of Individual User Navigation Patterns. In *Proceedings of Workshop on Individual Differences in Adaptive Hypermedia*, 40-49.

Juvina, I. & Herder, E. (2005). The Impact of Link Suggestions on User Navigation and User Perception, In *Proceedings of the 10th International Conference on User Modeling*.

Kitajima, M., Blackmon, M. H., & Polson, P.G. (2000). A Comprehension-based Model of Web Navigation and Its Application to Web Usability Analysis. *People and Computers XIV,* 357-373, Springer.

Liu, H. & Kešelj, V. (2007). Combined mining of Web server logs and web contents for classifying user navigation patterns and predicting users' future requests. *Data & Knowledge Engineering*, 61(2), pp. 304-330.

Mobasher, B. & Anand, S. S. (2005). *Intelligent Techniques for Web Personalization*. Lecture Notes in Artificial Intelligence. Springer.

Nakayama, T., Kato, H. & Yamane, Y. (2000). Discovering the gap between web site designers' expectations and users' behavior. In *Proceedings of the Ninth International World Wide Web Conference*.

Nasraoui, O., Cardona, C., Rojas, C. & Gonzalez, F. (2003). Mining Evolving User Profiles in Noisy Web Clickstream Data with a Scalable Immune System Clustering Algorithm. In *Workshop Notes of WEBKDD 2003: Web Mining as Premise to Effective and Intelligent Web Applications*, 71-81.

Paik, H.Y. Benatallah, B. & Hamadi, R. (2002). Dynamic Restructuring of E-Catalog Communities Based on User Interaction Patterns. *World Wide Web: Internet and Web Information Systems*, 5(4), pp. 325-366.

Pirolli, P., & Fu, W.-T. (2003). SNIF-ACT: A Model of Information Foraging on the World Wide Web. In *Proceedings of User Modeling 2003*.

Pirolli, P. & Card, S. K. (1995). Information foraging in information access environments. In *Proceedings of the CHI '95, ACM Conference on Human Factors in Software*, 51–58.

Missaoui, R., Valtchev, P., Djeraba, C. & Adda, M. (2007). Toward Recommendation Based on Ontology-Powered Web-Usage Mining. *IEEE Internet Computing*, 11(4), pp. 45-52.

Spiliopoulou, M. & Faulstich, L. C. (1998). Wum: A web utilization miner. In *Proceedings of EDBT Workshop WebDB98*.

Spiliopoulou, M., Faulstich, L. C. & Wilkler, K. (1999). A data miner analyzing the navigational behaviour of web users. In *Proceedings of the Workshop on Machine Learning in User Modeling of the ACAI99*.

Srikant, R. & Yang, Y. (2001). Mining Web Logs to Improve Website Organization. In *Proceedings of the 10th International World Wide Web Conference*, 430-437.

Srivastava, J., Cooley, R., Deshpande, M. & Tan, P. (2000). Web usage mining: Discovery and applications of usage patterns from web data. *SIGKDD Explorations*, 1(2), 1–12.

Tauscher, L. & Greenberg, S. (1997). How people revisit web pages: Empirical findings and implications for the design of history systems. *International Journal of Human-Computer Studies*, *47*, 97–137.

Yeung, P. C. K., Charles L. A., Clarke, C. L. A. & Büttcher, S. (2007). Improving retrieval accuracy by weighting document types with clickthrough data. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval.*

KEY TERMS

Adaptive Hypermedia System: It is an alternative to the traditional "one-size-fits-all" approach in the

development of hypermedia systems. Adaptive hypermedia systems build a model of the goals, preferences and knowledge of each individual user, and use this model throughout the interaction with the user, in order to adapt the hypertext to the needs of that user.

Information Foraging Theory: A general term for modeling a human searching for and using information in terms of biological models of animals foraging for food.

Information Scent: Information scent is a term used to describe how people evaluate the options they have when they are looking for information on a site. When presented with a list of options users will choose the option that gives them the clearest indication (or strongest scent) that it will step them closer to the information they require.

Link Recommendation: Based on users' previous browsing history, web links that the users may be interested in next are recommended.

Ontology: specification of a conceptualization of a knowledge domain. An ontology is a controlled vocabulary that describes objects and the relations between them in a formal way, and has a grammar for using the vocabulary terms to express something meaningful within a specified domain of interest.

Web Server Log: A file (or several files) automatically created and maintained by a web server recording all the requests for files made to the server and the result of that request.

Web Usage Mining: Extract useful access pattern information from massive amounts of web log data for the purpose of web site personalization and organization.

Web Mining in Thematic Search Engines

Massimiliano Caramia
University of Rome "Tor Vergata," Italy

Giovanni Felici
Istituto di Analisi dei Sistemi ed Informatica IASI-CNR, Italy

INTRODUCTION

In the present chapter we report on some extensions on the work presented in the first edition of the Encyclopedia of Data Mining. In Caramia and Felici (2005) we have described a method based on clustering and a heuristic search method- based on a genetic algorithm - to extract pages with relevant information for a specific user query in a thematic search engine. Starting from these results we have extended the research work trying to match some issues related to the semantic aspects of the search, focusing on the keywords that are used to establish the similarity among the pages that result from the query. Complete details on this method, here omitted for brevity, can be found in Caramia and Felici (2006).

Search engines technologies remain a strong research topic, as new problems and new demands from the market and the users arise. The process of switching from *quantity* (maintaining and indexing large databases of web pages and quickly select pages matching some criterion) to *quality* (identifying pages with a high quality for the user), already highlighted in Caramia and Felici (2005), has not been interrupted, but has gained further energy, being motivated by the natural evolution of the internet users, more selective in their choice of the search tool and willing to pay the price of providing extra feedback to the system and wait more time to have their queries better matched. In this framework, several have considered the use of data mining and optimization techniques, that are often referred to as *web mining* (for a recent bibliography on this topic see, e.g., Getoor, Senator, Domingos, and Faloutsos, 2003 and Zaïane, Srivastava, Spiliopoulou, and Masand, 2002).

The work described in this chapter is bases on clustering techniques to identify, in the set of pages resulting from a simple query, subsets that are homogeneous with respect to a vectorization based on *context* or *profile*; then, a number of small and potentially good subsets of pages is constructed, extracting from each cluster the pages with higher scores. Operating on these subsets with a genetic algorithm, a subset with a good overall score and a high internal dissimilarity is identified. A related problem is then considered: the selection of a subset of pages that are compliant with the search keywords, but that also are characterized by the fact that they share a large subset of words different from the search keywords. This characteristic represents a sort of semantic connection of these pages that may be of use to spot some particular aspects of the information present in the pages. Such a task is accomplished by the construction of a special graph, whose maximum-weight clique and k-densest subgraph should represent the page subsets with the desired properties.

In the following we summarize the main background topics and provide a synthetic description of the methods. Interested readers may find additional information in Caramia and Felici (2004), Caramia and Felici (2005), and Caramia and Felici (2006).

BACKGROUND

Let P be a set of web pages, and indicate with $p \in P$ a page in that set. Now assume that P is the result of a standard query to a database of pages, and thus represents a set of pages that satisfy some conditions expressed by the user. Each page $p \in P$ is associated with a score, based on the query that generated P, that would determine the order by which the pages are presented to the user who submits the query. The role of this ordering is crucial for the quality of the search: in fact, if the dimension of P is relevant, the probability that the user considers a page p strongly decreases as the position of p in the order increases. This may lead to two major drawbacks: the pages in the first positions may be very similar (or even equal) to each other; pages that do not have a very high score but are representative of some aspect of set P may appear in

a very low position in the ordering, with a negligible chance of being looked at by the user.

Our method tries to overcome both drawbacks, focusing on the selection, from the initial set *P*, of a small set of pages with a high score and sufficiently different one from each other. A condition needed to apply our approach is the availability of additional information from the user, who indicates a search context (a general topic to which the search is referred to, that is not necessarily linked with the search keywords that generated the set *P*), and a user profile (a subjective identification of the user, that may be either provided directly by choosing amongst a set of predefined profiles, or extracted from the pages that have been visited more recently by that user).

MAIN THRUST OF THE CHAPTER

The basic idea of the method is to use the information conveyed by the search context or the user profile to analyze the structure of *P* and determine in it an optimal small subset that better represents all the information available. This is done in three steps. First, the search context and the user profile are used to extract a finite set of significant words or page characteristics that is used to create, from all pages in *P*, a vector of characteristics (page vectorization). Such vectorization represents a particular way of "looking" at the page, specific of each context/profile, and will constitute the ground on which the following steps are based.

Second, the vectorized pages are analyzed by a *clustering algorithm* that partitions them into subsets of similar pages. This induces a two-dimensional ordering on the pages, as each page *p* can now be ordered according to the original score within its cluster. At this point the objective is to provide the user with a reduced list that takes into account the structure identified by the clusters and the original score function.

This is done in the third step, where a *genetic algorithm* works on the pages that have higher score in each cluster to produce a subset of them that are sufficiently heterogeneous and of good values for the original score. In the following we describe the three steps in detail.

We then report on the technique proposed to select pages that, beside being relevant for the search, are induced by a set of strongly connected words, with a proper definition of connection; in addition, we are inclined to select in this set those words that have a high degree of similarity with the search keywords, to enhance the significance of the induced pages. This problem is formulated as a maximum-weight clique problem on a graph whose nodes are associated with the initial set of words and whose arcs convey a weight based on the cardinality of page subsets associated with the word nodes.

Page Vectorization

The first step of the method is the representation of each page that has been acquired by a vector of finite dimension *m*, where each component represents a measure of some characteristic of the page (page vectorization). Clearly, such representation is crucial for the success of the method; all the information of a page that is not maintained in this step will be lost for further treatment. For this reason it is very important to stress the thematic nature of the vectorization process, where only the information that appears to be relevant for a context or a profile is effectively kept for future use. In the most plain setting, each component of the vector is the number of occurrences of a particular word; one may also consider other measurable characteristics that are not specifically linked with the words that are contained in the page, such as the presence of pictures, tables, banners and so on. As mentioned above, the vectorization is based on one context, or one profile, chosen by the user. One may then assume that, for each of the contextes/profiles that have been implemented in the search engine, a list of words that are relevant to that context/profile is available and a related vectorization of the page is stored. In Caramia and Felici (2005) we propose two methods to determine the initial list of words.

Page Clustering

There has been extensive research on how to improve retrieval results by employing clustering techniques. In several studies the strategy was to build a clustering of the entire document collection and then match the query to the cluster centroids (see, e.g., Willet, 1988). More recently, clustering has been used for helping the user in browsing a collection of documents and in organizing the results returned by a search engine (Leuski, 2001, and Zamir, Etzioni, Madani, and Karp, 1997), or by a metasearch engine (Zamir and Etzioni, 1999) in

response to a user query. In Koller and Sahami (1997) document clustering has also been used to automatically generate hierarchical clusters of documents.

Document clustering in information retrieval is usually dealt with agglomerative hierarchical clustering algorithms (see, e.g., Jain, Murty and Flynn, 1999) or *k*-means algorithm (see, e.g., Dubes and Jain, 1998). While agglomerative hierarchical clustering algorithms are very slow when applied to large document databases (Zamir and Etzioni, 1998) (single link and group average methods take $O(|P|^2)$ time, complete link method take $O(|P|^3)$ time), *k*-means is much faster (its execution time is $O(k \cdot |P|)$). We have tested different methods and confirm that *k*-means clustering performs well in limited computing times – a must for this type of applications, where both the number of pages and the dimension of the vectors may be large.

The Genetic Algorithm

Genetic algorithms have been implemented efficiently in information retrieval by several researchers. Chen (1995) used genetic algorithms to optimize keywords that were used to suggest relevant documents. Amongst others, Kraft, Petry, Buckles and Sadavisan (1997), Sanchez and Pierre (1994), presented several approaches to enhance the query description based on genetic algorithms. In Boughanem, Chrisment and Tamine (1999) a genetic algorithm was deployed to find an optimal set of documents that best match the user's need. In Horng and Yeh (2000) a method for extracting keywords from documents and assigning them weights was proposed.

Our aim is to select a small subset *P'* of the original set of pages *P* for which the sum of the scores is large but also where the similarity amongst the selected page is restrained. We select such a subset using a genetic algorithm (GA). Several reasons motivate this choice. First, the use of metaheuristic techniques is well established in optimization problems where the objective function and the constraints do not have a simple mathematical formulation. Second, we have to determine a good solution in a small computing time, where the dimension of the problem may be significantly large. Third, the structure of our problem is straightforward representable by the data structure commonly used by a GA (see, e.g., Goldberg, 1989).

Starting from the clusters obtained, we define the chromosomes of the initial population as subsets of pages of bounded cardinality (in the GA terminology, a page is a *gene*). The genetic algorithm works on the initial population ending up with a representative subset of pages to present to the user. The idea is to start the genetic evolution with a population that is already smaller than the initial set of pages *P*. Each chromosome is created by picking a page from each cluster, starting with the ones with higher score. Thus, the first chromosome created will contain the pages with the highest score in each cluster, the second chromosome will contain the second best, and so on. If the cardinality of a cluster is smaller than the number of chromosomes to be created, then that cluster will not be represented in each chromosome, while other clusters with higher cardinality may have more that one page representing them in some chromosome. We indicate with *dc* the number of pages included in each chromosome in the initial population, and with *nc* the number of chromosomes. The population will thus contain $np = dc \cdot nc$ pages.

The fitness function computed for each chromosome is expressed as a positive value that is higher for "better" chromosome, and is thus to be maximized. It is composed of three terms. The first is the sum of the score of the pages in chromosome *C*, i.e.,

$$t_1(C) = \sum_{p_i \in C} score(p_i)$$

where $score(p_i)$ is the original score given to page p_i as previously described. This term considers the positive effect of having as many pages with high score as possible in a chromosome, but would also reward chromosomes with many pages regardless of their. This drawback is balanced with the second term of the fitness function. Let *ID* be such ideal dimension; the ratio $t_2(C) = np / \text{abs}(|C|-ID)+1$ constitutes the second term of the fitness function: it reaches its maximum *np* when the dimension of *C* is exactly equal to the ideal dimension *ID*, and rapidly decreases when the number of pages contained in chromosome *C* is smaller or greater than *ID*.

The chromosomes that are present in the initial population are characterized by the highest possible variability as far as the clusters to which the pages belong are concerned. The evolution of the population may alter this characteristic, creating chromosomes with high fitness where the pages belong to the same cluster and are very similar to each other. Moreover, the fact that pages belonging to different clusters are

different in the vectorized space may not be guaranteed, as it depends both on the nature of the data and on the quality of the initial clustering process. For this reason, we introduce in the fitness function a third term, that measures directly the overall dissimilarity of the pages in the chromosome. Let $D(p_i, p_j)$ be the Euclidean distance of the vectors representing pages p_i and p_j. Then, $t_3(C)=\sum_{p_i,p_j\in C, p_i\neq p_j} D(p_i, p_j)$ is the sum of the distances between the pairs of pages in chromosome C and measures the total variability expressed by C. The final form of the fitness function for chromosome C is then $ff(C)=\alpha\cdot t_1(C)+\beta\cdot t_2(C)+\gamma\cdot t_3(C)$ where α, β and γ are parameters that depend on the magnitude of the initial score and of the vectors that represent the pages. In particular, α, β and γ are chosen so as the contributions given by $t_1(C)$, $t_2(C)$ and $t_3(C)$ are balanced. Additionally, they may be tuned to express the relevance attributed to the different aspects represented by the three terms. The goal of the GA is to find, by means of the genetic operators, a chromosome C^* such that $ff(C^*)=\max_{C=1,\ldots,nc} ff(C)$.

The Search for Semantically Connected Pages

The identification of semantically connected subsets of pages intends to overcome the typical drawbacks that a user encounters when he/she is submitted a set of pages selected according to some score function that is added on the single pages of the set rather than computed on the sets as a whole.

The main computational step to solve the problem is the identification of subgraphs of a given graph G with certain properties, namely some variants of two strongly related problems: the maximum-weight clique and the k-densest subgraph. A clique in a graph is a node induced subgraph where each couple of nodes is connected by an edge. The maximum-weight clique problem consists in finding a clique of largest weight in a graph. The densest k-subgraph problem amounts to the selection of the subgraph of dimension k with the minimum number of edges. Graph G is such that there is a node v for each word w with weight $a_v=\Sigma_{p\in P} M_{w,p}$, $M_{i,j}$ represents the number of times word w_j appears in page p_i, and an edge is present between each pair of nodes (u,v) with weight a_{uv} proportional to the inverse of the similarity between the words associated with u and v. In Caramia and Felici (2006) we proposed an original solution algorithm that is able to guarantee good performance in very little time and can find the optimal solution to the problem when the graph is of reasonable dimension.

FUTURE TRENDS

The application of sophisticated data analysis and data mining techniques to the search of information on the web is a field that receives increasing interest from both research and industry. The strategic importance of such tools should not be underestimated, as the amount of information keeps increasing while the user time available for searching is not increasing. Such trend motivates the effort of research to produce tools that help in improving web search results.

CONCLUSION

Experimental results conducted with the above described method have shown its effectiveness in the selection of small subsets of pages of good quality, where quality is not considered as a simple sum of the scores of each page but as a global characteristic of the subset. The current implementations of the GA and of the clustering algorithm converge fast to good solutions for data sets of realistic dimensions. Future research in the search for semantic connected pages will include the tuning of the parameters in the algorithms used to identify the subset of words, and a deeper analysis of the interactions between computational time and solution quality to possibly improve the quality of the method with ad-hoc heuristic strategies.

REFERENCES

Boughanem, M., Chrisment, C., & Tamine, L. (1999). Genetic Approach to Query Space Exploration. *Information Retrieval*, 1,175-192.

Chen, H. (1995). Machine Learning for Information Retrieval: Neural Networks, Symbolic Learning, and Genetic Algorithms. *Journal of the American Society for Information Science*, 46 (3), 194-216.

Caramia, M., & Felici, G. (2004). Improving Search Results with Data Mining in a Thematic Search Engine, Computers and Operations Research 31, 2387-2404.

Caramia, M., & Felici, G. (2005). Data Mining in a Web Search Engine, Encyclopedia of Data Warehousing and Mining, 2005.

Caramia, M., & Felici, G. (2006). Mining Relevant Information on the Web: A Clique Based Approach, *International Journal on Production Research*, 44, 2771-2787.

Dubes, R.C., & Jain, A. K. (1988). *Algorithms for Clustering Data*. Prentice Hall.

Getoor, L., Senator, T.E., Domingos, P., & Faloutsos, C. (Editors) (2003). 9*th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Washington, DC, USA.

Goldberg, E.D. (1999). *Genetic Algorithms in Search Optimization & Machine Learning*. Addison-Wesley Publishing Company.

Horng, J.T., & Yeh C.C. (2000). Applying genetic algorithms to query optimization in document retrieval. *Information Processing and Management*, 36, 737-759.

Jain, A.K, Murty, M.N., & Flynn, P.J. (1999). Data Clustering: a Review. *ACM Computing Surveys*, 31 (3), 264-323.

Koller, D., & Sahami, M. (1997). Hierarchically classifying documents using very few words. *Proc. of the* 14*th International Conference on Machine Learning*. Nashville, Tennessee, 170-178.

Kraft, D.H., Petry, F.E., Buckles, B.P., & Sadavisan, T. (1997). Genetic Algorithms for Query Optimization in Information Retrieval: Relevance Feedback. In Sanchez E., Zadeh L.A., Shibata T. (eds.), *Genetic Algorithms and Fuzzy Logic Systems, Soft Computing Perspectives*. World Scentific, 155-173.

Leuski, A. (2001). Evaluating Document Clustering for Interactive Information Retrieval. *Proc. of the* 2001 *ACM CIKM International Conference on Information and Knowledge Management*. Atlanta, Georgia, USA, 33-44.

Sanchez, E., & Pierre, Ph. (1994). Fuzzy Logic and Genetic Algorithms in Information Retrieval. *Proceedings of the 3rd International Conference on Fuzzy Logic, Neural Net and Soft Computing*. Iizuka, Japan, 29-35.

Zaïane, O.R., Srivastava, J., Spiliopoulou, M., & Masand, B.M. (Editors) (2002). *International Workshop in Mining Web Data for Discovering Usage Patterns and Profiles*, Edmonton, Canada.

Zamir, O., & Etzioni, O. (1999). Grouper: a dynamic clustering interface to web search results. *Proceedings of the* 8*th International Conference on World Wide Web*. Toronto, Canada, 1361-1374.

Zamir, O., & Etzioni, O. (1998). Web Document Clustering: A Feasibility Demonstration. *Proc. of* 19*th international ACM SIGIR conference on research and development in information retrieval*, SIGIR 98. Melbourne, Australia, 46-54.

Zamir, O., Etzioni, O., Madani, O., & Karp, R.M. (1997). Fast and Intuitive Clustering of Web Documents. Proceedings of the 3rd Interational Conference on Knowledge Discovery and Databases, KDD '97. Newport Beach, California, 287-290.

KEY TERMS

Clustering: partitioning a data set into subsets (clusters), so that the data in each subset share some common trait.

Genetic Algorithm: heuristic optimization algorithm based on the concept of biological evolution.

Page Score: numeric value that measures how well a single page matches a given query. Higher score would imply a better matching.

Search Engine: software that builds a database of web pages, applies queries to it and returns results.

Thematic Search Engine: search engine devoted to the construction and management of a database of web pages that pertain to a limited subset of the knowledge or of the web users.

Vectorization: the representation of objects in a class by a finite set of measures defined on the objects.

Web Page: basic unit of information that is visualized on the web

Web Mining Overview

Bamshad Mobasher
DePaul University, USA

INTRODUCTION

In the span of a decade, the World Wide Web has been transformed from a tool for information sharing among researchers into an indispensable part of everyday activities. This transformation has been characterized by an explosion of heterogeneous data and information available electronically, as well as increasingly complex applications driving a variety of systems for content management, e-commerce, e-learning, collaboration, and other Web services. This tremendous growth, in turn, has necessitated the development of more intelligent tools for end users as well as information providers in order to more effectively extract relevant information or to discover actionable knowledge.

From its very beginning, the potential of extracting valuable knowledge from the Web has been quite evident. Web mining (i.e. the application of data mining techniques to extract knowledge from Web content, structure, and usage) is the collection of technologies to fulfill this potential.

In this article, we will summarize briefly each of the three primary areas of Web mining—Web usage mining, Web content mining, and Web structure mining—and discuss some of the primary applications in each area.

BACKGROUND

Knowledge discovery on and from the Web has been characterized by four different but related types of activities (Kosala & Blockeel, 2000):

1. **Resource Discovery:** Locating unfamiliar documents and services on the Web.
2. **Information Extraction:** Extracting automatically specific information from newly discovered Web resources.
3. **Generalization:** Uncovering general patterns at individual Web sites or across multiple sites.
4. **Personalization:** Presentation of the information requested by an end user of the Web.

The goal of Web mining is to discover global as well as local structures, models, patterns, or relations within and between Web pages. The research and practice in Web mining has evolved over the years from a process-centric view, which defined Web mining as a sequence of tasks (Etzioni, 1996), to a data-centric view, which defined Web mining in terms of the types of Web data that were being used in the mining process (Cooley et al., 1997).

MAIN THRUST

The evolution of Web mining as a discipline has been characterized by a number of efforts to define and expand its underlying components and processes (Cooley et al., 1997; Kosla & Blockeel, 2000; Madria et al., 1999; Srivastava et al., 2002). These efforts have led to three commonly distinguished areas of Web mining: Web usage mining, Web content mining, and Web structure mining.

Web Content Mining

Web content mining is the process of extracting useful information from the content of Web documents. Content data correspond to the collection of facts a Web page was designed to convey to the users. Web content mining can take advantage of the semi-structured nature of Web page text. The HTML tags or XML markup within Web pages bear information that concerns not only layout but also the logical structure and semantic content of documents.

Text mining and its application to Web content have been widely researched (Berry, 2003; Chakrabarti, 2000). Some of the research issues addressed in text mining are topic discovery, extracting association patterns, clustering of Web documents, and classification of Web pages. Research activities in this field generally

involve using techniques from other disciplines, such as information retrieval (IR), information extraction (IE), and natural language processing (NLP).

Web content mining can be used to detect co-occurrences of terms in texts (Chang et al., 2000). For example, co-occurrences of terms in newswire articles may show that gold frequently is mentioned together with copper when articles concern Canada, but together with silver when articles concern the US. Trends over time also may be discovered, indicating a surge or decline in interest in certain topics, such as programming languages like Java. Another application area is event detection, the identification of stories in continuous news streams that correspond to new or previously unidentified events.

A growing application of Web content mining is the automatic extraction of semantic relations and structures from the Web. This application is closely related to information extraction and ontology learning. Efforts in this area have included the use of hierarchical clustering algorithms on terms in order to create concept hierarchies (Clerkin et al., 2001), the use of formal concept analysis and association rule mining to learn generalized conceptual relations (Maedche & Staab, 2000; Stumme et al., 2000), and the automatic extraction of structured data records from semi-structured HTML pages (Liu, Chin & Ng, 2003). Often, the primary goal of such algorithms is to create a set of formally defined domain ontologies that represent precisely the Web site content and to allow for further reasoning. Common representation approaches are vector-space model (Loh et al., 2000), descriptive logics (i.e., DAML+OIL) (Horrocks, 2002), first order logic (Craven et al., 2000), relational models (Dai & Mobasher, 2002), and probabilistic relational models (Getoor et al., 2001).

Web Structure Mining

The structure of a typical Web graph consists of Web pages as nodes and hyperlinks as edges connecting between two related pages. Web structure mining can be regarded as the process of discovering structure information from the Web. This type of mining can be divided further into two kinds, based on the kind of structural data used (Srivastava et al., 2002); namely, hyperlinks or document structure. There has been a significant body of work on hyperlink analysis, of which Desikan et al. (2002) provide an up-to-date survey. The content within a Web page also can be organized in a tree-structured format, based on the various HTML and XML tags within the page. Mining efforts here have focused on automatically extracting document object model (DOM) structures out of documents (Moh et al., 2000) or on using the document structure to extract data records or semantic relations and concepts (Liu, Chin & Ng, 2003; Liu, Grossman & Zhai, 2003).

By far, the most prominent and widely accepted application of Web structure mining has been in Web information retrieval. For example, the Hyperlink Induced Topic Search (HITS) algorithm (Klienberg, 1998) analyzes hyperlink topology of the Web in order to discover authoritative information sources for a broad search topic. This information is found in authority pages, which are defined in relation to hubs as their counterparts: Hubs are Web pages that link to many related authorities; authorities are those pages that are linked by many good hubs. The hub and authority scores computed for each Web page indicate the extent to which the Web page serves as a hub pointing to good authority pages or as an authority on a topic pointed to by good hubs.

The search engine Google also owes its success to the PageRank algorithm, which is predicated on the assumption that the relevance of a page increases with the number of hyperlinks pointing to it from other pages and, in particular, of other relevant pages (Brin & Page, 1998). The key idea is that a page has a high rank, if it is pointed to by many highly ranked pages. So, the rank of a page depends upon the ranks of the pages pointing to it. This process is performed iteratively until the rank of all the pages is determined.

The hyperlink structure of the Web also has been used to automatically identify Web communities (Flake et al., 2000; Gibson et al., 1998). A Web community can be described as a collection of Web pages, such that each member node has more hyperlinks (in either direction) within the community than outside of the community.

An excellent overview of techniques, issues, and applications related to Web mining, in general, and to Web structure mining, in particular, is provided in Chakrabarti (2003).

Web Usage Mining

Web usage mining (Cooley et al., 1999; Srivastava et al., 2000) refers to the automatic discovery and analysis

of patterns in clickstream and associated data collected or generated as a result of user interactions with Web resources on one or more Web sites. The goal of Web usage mining is to capture, model, and analyze the behavioral patterns and profiles of users interacting with a Web site. The discovered patterns are usually represented as collections of pages, objects, or resources that are frequently accessed by groups of users with common needs or interests.

The primary data sources used in Web usage mining are log files automatically generated by Web and application servers. Additional data sources that also are essential for both data preparation and pattern discovery include the site files and meta-data, operational databases, application templates, and domain knowledge.

The overall Web usage mining process can be divided into three interdependent tasks: data preprocessing, pattern discovery, and pattern analysis or application. In the preprocessing stage, the clickstream data is cleaned and partitioned into a set of user transactions representing the activities of each user during different visits to the site. In the pattern discovery stage, statistical, database, and machine learning operations are performed to obtain possibly hidden patterns reflecting the typical behavior of users, as well as summary statistics on Web resources, sessions, and users. In the final stage of the process, the discovered patterns and statistics are further processed, filtered, and used as input to applications such as recommendation engines, visualization tools, and Web analytics and report generation tools.

For a full discussion of Web usage mining and its applications, see the article "Web Usage Mining" in the current volume (Mobasher, 2005).

FUTURE TRENDS

An important emerging area that holds particular promise is Semantic Web Mining (Berendt et al., 2002). Semantic Web mining aims at combining the two research areas: semantic Web and Web mining. The primary goal is to improve the results of Web mining by exploiting the new semantic structures on the Web. Furthermore, Web mining techniques can help to automatically build essential components of the Semantic Web by extracting useful patterns, structures, and semantic relations from existing Web resources (Berendt et al., 2004).

Other areas in which Web mining research and practice is likely to make substantial gains are Web information extraction, question-answering systems, and personalized search. Progress in the applications of natural language processing as well as increasing sophistication of machine learning and data mining techniques applied to Web content are likely to lead to the development of more effective tools for information foraging on the Web. Some recent advances in these areas have been highlighted in recent research activities (Mobasher et al. 2004; Muslea et al. 2004).

CONCLUSION

Web mining is the application of data mining techniques to extract knowledge from the content, structure, and usage of Web resources. With the continued growth of the Web as an information sources and as a medium for providing Web services, Web mining will continue to play an ever expanding and important role. The development and application of Web mining techniques in the context of Web content, Web usage, and Web structure data already have resulted in dramatic improvements in a variety of Web applications, from search engines, Web agents, and content managements systems to Web analytics and personalization services. A focus on techniques and architectures for more effective integration and mining of content, usage, and structure data from different sources is likely to lead to the next generation of more useful and more intelligent applications.

REFERENCES

Berendt, B., Hotho, A., Mladenic, D., van Someren, M., & Spiliopoulou, M. (2004). Web mining: From Web to semantic Web. *Lecture Notes in Computer Science, Vol. 3209*. Heidelberg, Germany: Springer-Verlag.

Berendt, B., Hotho, A., & Stumme, G. (2002). Towards semantic Web mining. *Proceedings of the First International Semantic Web Conference (ISWC02)*, Sardinia, Italy.

Berry, M. (2003). *Survey of text mining: Clustering, classification, and retrieval*. Heidelberg, Germany: Springer-Verlag.

Brin, S., & Page, L. (1998). The anatomy of a large-scale hyper-textual Web search engine. *Proceedings of the 7th International World Wide Web Conference*, Brisbane, Australia.

Chakrabarti, S. (2000). Data mining for hypertext: A tutorial survey. *SIGKDD Explorations, 1*(2), 1-11.

Chakrabarti, S. (2003). *Mining the Web: Discovering knowledge from hypertext data*. San Francisco, CA: Morgan Kaufmann.

Chang, G., Healey, M.J., McHugh, J.A.M., & Wang, J.T.L. (2001). *Mining the World Wide Web: An information search approach*. Boston: Kluwer Academic Publishers.

Clerkin, P., Cunningham, P., & Hayes, C. (2001). Ontology discovery for the semantic Web using hierarchical clustering. *Proceedings of the Semantic Web Mining Workshop at ECML/PKDD-2001*, Freiburg, Germany.

Cooley, R., Mobasher, B., & Srivastava, J. (1997). Web mining: Information and pattern discovery on the World Wide Web. *Proceedings of the 9th IEEE International Conference on Tools With Artificial Intelligence (ICTAI '97)*, Newport Beach, California.

Cooley, R., Mobasher, B., & Srivastava, J. (1999). Data preparation for mining World Wide Web browsing patterns. *Knowledge and Information Systems*, *1*(1), 5-32.

Craven, M. et al. (2000). Learning to construct knowledge bases from the World Wide Web. *Artificial Intelligence, 118*(1-2), 69-113.

Dai, H., & Mobasher, B. (2002). Using ontologies to discover domain-level Web usage profiles. *Proceedings of the 2nd Semantic Web Mining Workshop at ECML/PKDD 2002*, Helsinki, Finland.

Desikan, P., Srivastava, J., Kumar, V., & Tan, P.-N. (2002). *Hyperlink analysis: Techniques and applications* [technical report]. Minneapolis, MN: Army High Performance Computing Center.

Etzioni, O. (1996). The World Wide Web: Quagmire or gold mine. *Communications of the ACM, 39*(11), 65-68.

Flake, G.W., Lawrence, S., & Giles, C.L. (2000). Efficient identification of Web communities. *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2000)*, Boston.

Getoor, L., Friedman, N., Koller, D., & Taskar, B. (2001). Learning probabilistic models of relational structure. *Proceedings of the 18th International Conference on Machine Learning*, Williamstown, MA.

Gibson, D., Kleinberg, J., & Raghavan, P. (1998). Inferring Web communities from link topology. *Proceedings of the Ninth ACM Conference on Hypertext and Hypermedia*, Pittsburgh, Pennsylvania.

Horrocks, I. (2002). DAML+OIL: A description logic for the semantic Web. *IEEE Data Engineering Bulletin, 25*(1), 4-9.

Kleinberg, M. (1998). Authoritative sources in hyperlinked environment. *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms,* San Francisco, California.

Kosala, R., & Blockeel, H. (2000). Web mining research: A survey. *SIGKDD Explorations, 2*(1), 1-15.

Liu, B., Chin, C.W., & Ng, H.T. (2003). Mining topic-specific concepts and definitions on the Web. *Proceedings of the Twelfth International World Wide Web Conference (WWW-2003)*, Budapest, Hungary.

Liu, B., Grossman, R., & Zhai, Y. (2003). Mining data records in Web pages. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD-2003)*, Washington, D.C.

Loh, S., Wives, L.K., & de Oliveira, J.P. (2000). Concept-based knowledge discovery in texts extracted from the Web. *SIGKDD Explorations*, *2*(1), 29-39.

Madria, S., Bhowmick, S., Ng, W.K., & Lim, E.-P. (1999). Research issues in Web data mining. *Proceedings of Data Warehousing and Knowledge Discovery, First International Conference*, Florence, Italy.

Maedche, A., & Staab, S. (2000). Discovering conceptual relations from text. *Proceedings of the European Conference on Artificial Intelligence (ECAI00)*, Berlin, Germany.

Mobasher, B. (2005). Web usage mining. In J. Wang (Ed.), *Web usage mining data preparation*. Hershey, PA: Idea Group Publishing.

Mobasher, B., Liu, B., Masand, B., & Nasraoui, O. (2004). Web mining and Web usage analysis. *Proceedings of the 6th WebKDD workshop at the 2004 ACM SIGKKDD Conference*, Seattle, Washington.

Moh, C-H., Lim, E-P., & Ng, W.K. (2000). DTD-miner: A tool for mining DTD from XML documents. *Proceedings of Second International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems*, San Jose, California.

Muslea, I. et al. (2004). *Proceedings of the AAAI 2004 Workshop on Adaptive Text Extraction and Mining, ATEM-2004*, San Jose, California.

Srivastava, J., Cooley, R., Deshpande, M., & Tan, P. (2000). Web usage mining: Discovery and applications of usage patterns from Web data. *SIGKDD Explorations, 1*(2), 12-23.

Srivastava, J., Desikan, P., & Kumar, V. (2002). Web mining—Accomplishments and future directions. *Proceedings of the National Science Foundation Workshop on Next Generation DataMining (NGDM'02)*, Baltimore, Maryland.

Stumme, G., Taouil, R., Bastide, Y., Pasquier, N., & Lakhal, L. (2000). Fast computation of concept lattices using data mining techniques. *Proceedings of the Knowledge Representation Meets Databases Conference (KRDB00)*, Berlin, Germany.

KEY TERMS

Hubs and Authorities: Hubs and authorities are Web pages defined by a mutually reinforcing relationship with respect to their hyperlink structure. Hubs are Web pages that link to many related authorities; authorities are those pages that are linked to by many good hubs.

Hyperlink: A hyperlink is a structural unit that connects a Web page to a different location, either within the same Web page or to a different Web page. A hyperlink that connects to a different part of the same page is called an intra-document hyperlink, and a hyperlink that connects two different pages is called an inter-document hyperlink.

Web Community: A Web community can be described as a collection of Web pages, such that each member node has more hyperlinks (in either direction) within the community than outside of the community.

Web Content Mining: The process of extracting useful information from the contents of Web documents. Content data corresponds to the collection of facts that a Web page was designed to convey to users. It may consist of unstructured or semi-structured text, images, audio, video, or structured records, such as lists and tables.

Web Mining: The application of data-mining techniques to extract knowledge from the content, structure, and usage of Web resources. It is generally subdivided into three independent but related areas: Web usage mining, Web content mining, and Web structure mining.

Web Structure Mining: Web structure mining can be regarded as the process of discovering structure information from the Web. This type of mining can be divided further into two kinds, based on the kind of structural data used: hyperlinks connecting Web pages and the document structure in semi-structured Web pages.

Web Usage Mining: The automatic discovery and analysis of patterns in clickstream and associated data collected or generated as a result of user interactions with Web resources on one or more Web sites.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 523-528, copyright 2005 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

Web Page Extension of Data Warehouses

Anthony Scime
State University of New York College Brockport, USA

INTRODUCTION

Data warehouses are constructed to provide valuable and current information for decision-making. Typically this information is derived from the organization's functional databases. The data warehouse is then providing a consolidated, convenient source of data for the decision-maker. However, the available organizational information may not be sufficient to come to a decision. Information external to the organization is also often necessary for management to arrive at strategic decisions. Such external information may be available on the World Wide Web; and when added to the data warehouse extends decision-making power.

The Web can be considered as a large repository of data. This data is on the whole unstructured and must be gathered and extracted to be made into something valuable for the organizational decision maker. To gather this data and place it into the organization's data warehouse requires an understanding of the data warehouse metadata and the use of Web mining techniques (Laware, 2005).

Typically when conducting a search on the Web, a user initiates the search by using a search engine to find documents that refer to the desired subject. This requires the user to define the domain of interest as a keyword or a collection of keywords that can be processed by the search engine. The searcher may not know how to break the domain down, thus limiting the search to the domain name. However, even given the ability to break down the domain and conduct a search, the search results have two significant problems. One, Web searches return information about a very large number of documents. Two, much of the returned information may be marginally relevant or completely irrelevant to the domain. The decision maker may not have time to sift through results to find the meaningful information.

A data warehouse that has already found domain relevant Web pages can relieve the decision maker from having to decide on search keywords and having to determine the relevant documents from those found in a search. Such a data warehouse requires previously conducted searches to add Web information.

BACKGROUND

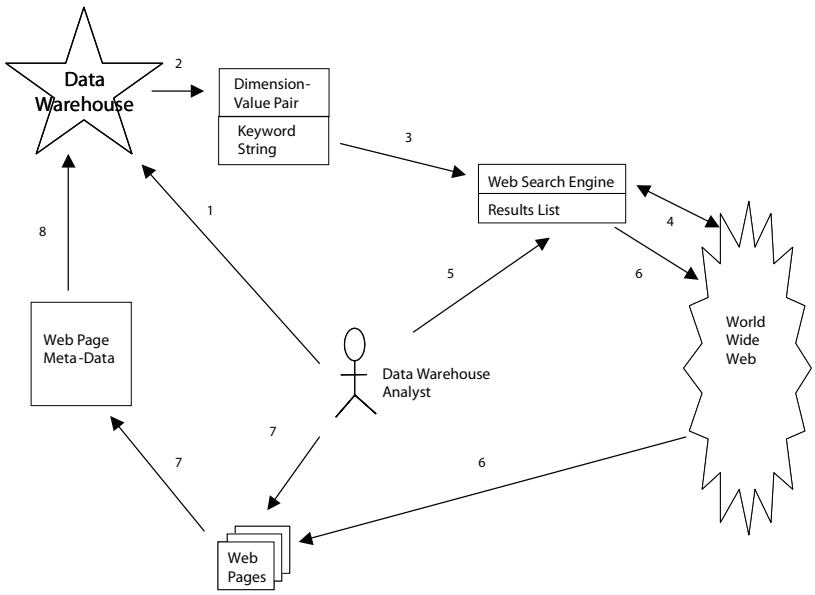
To provide an information source within an organization's knowledge management system, database structure has been overlaid on documents (Liongosari, Dempski, & Swaminathan, 1999). This knowledge base provides a source for obtaining organizational knowledge. Data warehouses also can be populated in Web-based interoperational environments created between companies (Triantafillakis, Kanellis & Martakos, 2004). This extends knowledge between cooperating businesses. However, these systems do not explore the public documents available on the Web.

Systems have been designed to extract relevant information from unstructured sources such as the Web. The Topicshop system allows users to gather, evaluate, and organize collections of Web sites (Amento, Terveen, Hill, Hix, & Schulman, 2003). Using topic discovery techniques Usenet news searching can be personalized to categorize contents and optimise delivery contents for review (Manco, Ortale & Tagarelli, 2005). Specialized search engines and indexes have been developed for many domains (Leake & Scherle, 2001). Search engines have been developed to combine the efforts of other engines and select the best search engine for a domain (Meng, Wu, Yu, & Li, 2001). However, these approaches do not organize the search results into accessible, meaningful, searchable data.

Web search queries can be related to each other by the results returned (Wu & Crestani, 2004; Glance, 2000). This knowledge of common results to different queries can assist a new searcher in finding desired information. However, it assumes domain knowledge sufficient to develop a query with keywords, and does not provide corresponding organizational knowledge.

Some Web search engines find information by categorizing the pages in their indexes. One of the first to create a structure as part of their Web index was

Figure 1. Data warehouse Web extension architecture



Yahoo! (http://www.yahoo.com). Yahoo! has developed a hierarchy of documents, which is designed to help users find information faster. This hierarchy acts as a taxonomy of the domain. Yahoo! helps by directing the searcher through the domain. Again, there is no organizational knowledge to put the Web pages into a local context, so the documents must be accessed and assimilated by the searcher.

DynaCat provides knowledge-based, dynamic categorization of search results in the medical domain (Pratt, 1999). The domain of medical topics is established and matched to predefined query types. Retrieved documents from a medical database are then categorized according to the topics. Such systems use the domain as a starting point, but do not catalog the information and add it to an existing organized body of domain knowledge such as a data warehouse.

Web pages that contain multiple semi-structured records can be parsed and used to populate a relational database. Multiple semi-structured records are data about a subject that is typically composed of separate information instances organized individually, but generally in the same format. For example, a Web page of want ads or obituaries. The first step is to create an ontology of the general structure of the semi-structured data. The ontology is expressed as an Object-Relationship Model. This ontology is then used to define the parsing of the Web page. Parsing into records uses the HTML tags to determine the structure of the Web page, determining when a record starts and ends. The relational database structure is derived from the ontology. The system requires multiple records in the domain, with the Web page having a defined structure to delimit records. However, the Web pages must be given to the system, it cannot find Web pages, or determine if they belong to the domain (Embley et al., 1999).

The Web Ontology Extraction (WebOntEx) project semi-automatically determines ontologies that exist on the Web. These ontologies are domain specific and placed in a relational database schema. Using the belief that HTML tags typically highlight a Web page's concepts, concepts are extracted, by selecting some number of words after the tag as concepts. They are reviewed and may be selected to become entity sets, attributes or relationships in a domain relational database. The determination is based on the idea that nouns are possible entity and attribute types and verbs are possible relationship types. By analyzing a number of pages in a domain an ontology is developed within the relational database structure (Han & Elmasri, 2004). This system creates the database from Web page input, whereas an existing data warehouse needs only to be extended with Web available knowledge.

Web based catalogs are typically taxonomy-directed. A taxonomy-directed Web site has its contents organized in a searchable taxonomy, presenting the instances of a category in an established manner. DataRover is a system that automatically finds and extracts products from taxonomy-directed, online catalogs. It utilizes heuristics to turn the online catalogs into a database of categorized products (Davulcu, Koduri & Nagarajan, 2003). This system is good for structured data, but is not effective on unstructured, text data.

To find domain knowledge in large databases domain experts are queried as to the topics and subtopics of a domain creating an expert level taxonomy (Scime, 2000, 2003). This domain knowledge can be used to assist in restricting the search space. The results found are attached to the taxonomy and evaluated for validity; and create and extend the searchable data repository.

WEB SEARCH FOR WAREHOUSING

Experts within a domain of knowledge are familiar with the facts and the organization of the domain. In the warehouse design process, the analyst extracts from the expert the domain organization. This organization is the foundation for the warehouse structure and specifically the dimensions that represent the characteristics of the domain.

In the Web search process; the data warehouse analyst can use the warehouse dimensions as a starting point for finding more information on the World Wide Web. These dimensions are based on the needs of decision makers and the purpose of the warehouse. They represent the domain organization. The values that populate the dimensions are pieces of the knowledge about the warehouse's domain. These organizational and knowledge faucets can be combined to create a dimension-value pair, which is a special case of a taxonomy tree (Kerschberg, Kim & Scime, 2003; Scime & Kerschberg, 2003). This pair is then used as keywords to search the Web for additional information about the domain and this particular dimension value.

The pages retrieved as a result of dimension-value pair based Web searches are analyzed to determine relevancy. The meta-data of the relevant pages is added to the data warehouse as an extension of the dimension. Keeping the warehouse current with frequent Web searches keeps the knowledge fresh and allows decision makers access to the warehouse and Web knowledge in the domain.

WEB PAGE COLLECTION AND WAREHOUSE EXTENSION

The Data Warehouse Web Extension Architecture (Figure 1) shows the process for adding Web pages to a data warehouse.

1. **Select Dimensions:** The data warehouse analyst selects the dimension attributes that are likely to have relevant data about their values on the Web. For example, the dimension *city* would be chosen; as most cities have Web sites.
2. **Extract Dimension Value Pair:** As values are added to the selected dimensions the dimension label and value are extracted as a dimension-value pair and converted into a keyword string. The value *Buffalo* for the dimension *city* becomes the keyword string *city Buffalo*.
3. **Keyword String Query:** The keyword string is sent to a search engine (for example, Google).
4. **Search the World Wide Web:** The keyword string is used as a search engine query and the resulting hit lists containing Web page meta-data are returned. This meta-data typically includes page URL, title, and some summary information. In our example, the first result is the Home Page for the City of Buffalo in New York State. On the second page of results is the City of Buffalo, Minnesota.
5. **Review Results Lists:** The data warehouse analyst reviews the resulting hit list for possible relevant pages. Given that large number of hits (over 5 million for *city Buffalo*) the analyst must limit consideration of pages to a reasonable amount.
6. **Select Web Documents:** Web pages selected are those that may add knowledge to the data warehouse. This may be new knowledge or extensional knowledge to the warehouse. Because the analyst knows that the city of interest to the data warehouse is Buffalo, New York, he only considers the appropriate pages.
7. **Relevancy Review:** The analyst reviews the selected pages to ensure they are relevant to the intent of the warehouse attribute. The meta-data of the relevant Web pages is extracted during this

relevancy review. The meta-data includes the Web page URL, title, date retrieved, date created, and summary. This meta-data may come from the search engine results list. For the Buffalo home page this meta-data is found in Figure 2.

8. **Add Meta-Data:** The meta-data for the page is added as an extension to the data warehouse. This addition is added as an extension to the *city* dimension creating a snowflake-like schema for the data warehouse.

FUTURE TRENDS

There are two trends in data repositories that when combined will greatly enhance the ability to extend data warehouses with Web based information. The first is the movement to object-oriented databases (Ravat, Teste & Zurfluh, 1999). The other is the movement to the semantic Web (Engels & Lech, 2003).

Currently, modeling and implementation of databases uses the Entity-Relationship model. This model has difficulty in representing multidimensional data views common in today's data warehouses. The object-oriented paradigm provides increased modeling capabilities in the multidimensional environment (Trujillo, Palomar & Gómez, 2000). Furthermore, the object-oriented data warehouse can be organized as an ontology.

In the search engine of the future the linear index of Web pages will be replaced by an ontology. This ontology will be a semantic representation of the Web. Within the ontology the pages may be represented by keywords and will also have connections to other pages. These connections will be the relationships between the pages and may also be weighted. Investigation of an individual page's content, the inter-page hypertext links, the position of the page on its server, and search engine discovered relationships would create the ontology (Guha, McCool & Miller, 2003).

Matches will no longer be query keyword to index keyword, but a match of the data warehouse ontology to the search engine ontology. Rather than point-to-point matching, the query is the best fit of one multi-dimensional space upon another (Doan, Madhavan, Dhamankar, Domingos, & Halevy, 2003). The returned page locations are then more specific to the information need of the data warehouse.

CONCLUSION

The use of the Web to extend data warehouse knowledge about a domain provides the decision maker with more information than may otherwise be available from the organizational data sources used to populate the data warehouse. The Web pages referenced in the warehouse are derived from the currently available data and knowledge of the data warehouse structure.

The Web search process and the data warehouse analyst sifts through the external, distributed Web to find relevant pages. This Web generated knowledge is added to the data warehouse for decision maker consideration.

REFERENCES

Amento, B., Terveen, L., Hill, W., Hix, D., & Schulman, R. (2003). Experiments in social data mining: The TopicShop system. *ACM Transactions on Computer-Human Interaction*, *10*(1), 54-85.

Davulcu, H., Koduri, S., & Nagarajan, S. (2003). Datarover: A taxonomy-based crawler for automated data extraction from data-intensive Websites. *Proceedings of the Fifth ACM International Workshop on Web Information and Data Management* (pp. 9-14), New Orleans, Louisiana.

Doan, A., Madhavan, J., Dhamankar, R., Domingos, P., & Halevy, A. (2003). Learning to match ontologies on the semantic Web. *The International Journal on Very Large Data Bases*, *12*(4), 303-319.

Embley, D.W., Campbell, D.M., Jiang, Y.S., Liddle, S.W., Lonsdale, D.W., Ng, Y.K., et al., (1999). Conceptual-model-based data extraction from multiple-record Web pages. *Data & Knowledge Engineering*, *31*(3), 227-251.

Engels, R., & Lech, T. (2003). Generating ontologies for the semantic Web: OntoBuilder. In J. Davies, & F.D. Van Harmelem (Eds.), *Towards the semantic Web: Ontology-driven Knowledge management* (pp. 91-115). U.K.: John Wiley & Sons.

Glance, N.S. (2000). Community search assistant. *AAAI Workshop Technical Report of the Artificial Intelligence for Web Search Workshop* (pp. 29-34), Austin, Texas.

Guha, R., McCool, R., & Miller, E. (2003). Semantic search. *Proceedings of the Twelfth International Conference on the World Wide Web* (pp. 700-709), Budapest, Hungary.

Han, H. & Elmasri, R. (2004). Learning rules for conceptual structure on the Web. *Journal of Intelligent Information Systems*, *22*(3), 237-256.

Kerschberg, L., Kim, W. & Scime, A. (2003). A personalizable agent for semantic taxonomy-based Web search. In W. Truszkowski, C. Rouff, & M. Hinchey (Eds.), *Innovative concepts for agent-based systems*. *Lecture notes in artificial intelligence* 2564 (pp. 3-31). Heidelberg: Springer.

Laware, G. (2005). Metadata management: A requirement for Web warehousing and knowledge management. In A. Scime (Ed.), *Web mining: Applications and techniques* (pp. 1-26). Hershey: Idea Group Publishing.

Leake, D. B. & Scherle, R. (2001). Towards context-based search engine selection. *Proceedings of the 6th International Conference on Intelligent User Interfaces* (pp. 109-112), Santa Fe, New Mexico.

Liongosari, E.S., Dempski, K.L., & Swaminathan, K.S. (1999). In search of a new generation of knowledge management applications. *SIGGROUP Bulletin*, *20*(2), 60-63.

Manco, G., Ortale, R., & Tagarelli, A. (2005). The scent of a newsgroup: Providing personalized access to usenet sites through Web mining. In A. Scime (Ed.), *Web mining: Applications and techniques* (pp. 393-413). Hershey: Idea Group Publishing.

Meng, W., Wu, Z., Yu, C., & Li, Z. (2001). A highly scalable and effective method for metasearch. *ACM Transactions on Information Systems*, *19*(3), 310-335.

Pratt, W., Hearst, M., & Fagan, L. (1999). A knowledge-based approach to organizing retrieved documents. *AAAI-99: Proceedings of the Sixteenth National Conference on Artificial Intelligence* (pp. 80-85), Orlando, Florida.

Ravat, F., Teste, O., & Zurfluh, G. (1999). Towards data warehouse design. *Proceedings of the Eighth International Conference on Information and Knowledge Management* (pp. 359-366), Kansas City, Missouri.

Scime, A. (2000). Learning from the World Wide Web: Using organizational profiles in information searches. *Informing Science*, *3*(3), 135-143.

Scime, A. (2003). Web mining to create a domain specific Web portal database. In D. Taniar & J. Rahayu (Eds.), *Web-powered databases* (pp. 36-53). Hershey: Idea Group Publishing.

Scime, A. & Kerschberg, L. (2003). WebSifter: An ontological Web-mining agent for e-business. In R. Meersman, K. Aberer, & T. Dillon (Eds.), *Semantic issues in e-commerce systems* (pp. 187-201). The Netherlands: Kluwer Academic Publishers.

Triantafillakis, A., Kanellis, P., & Martakos, D. (2004). Data warehouse interoperability for the extended enterprise. *Journal of Database Management*, *15*(3), 73-83.

Trujillo, J., Palomar, M., & Gómez, J. (2000). The GOLD definition language (GDL): An object-oriented formal specification language for multidimensional databases. *Proceedings of the 2000 ACM Symposium on Applied Computing* (pp. 346-350), Como, Italy.

Wu, S. & Crestani, F. (2004). Shadow document methods of results merging. *Proceedings of the 2004 ACM Symposium on Applied Computing* (pp. 1067-1072), Nicosia, Cyprus.

KEY TERMS

Dimension: A category of information relevant to the decision making purpose of the data warehouse.

Domain: The area of interest for which a data warehouse was created.

Meta-Data: Data about data. In a database the attributes, relations, files, etc. have labels or names indicating the purpose of the attribute, relation, file, etc. These labels or names are meta-data.

Search Engine: A Web service that allows a user to find Web pages matching the user's selection of keywords.

Star Schema: The typical logical topology of a data warehouse; where a fact table occupies the center of the data warehouse and dimension tables are related to most fact table attributes.

Taxonomy Tree: A collection of related concepts organized in a tree structure where higher-level concepts are decomposed into lower-level concepts.

URL: The Uniform Resource Locator (URL) is the address of all Web pages, images, and other resources on the World Wide Web.

Web Page: A file that is on the Web and is accessible by its URL.

Web Site: A collection of Web pages located together on a Web server. Typically the pages of a Web site have a common focus and are connected by hyperlinks.

This work was previously published in Encyclopedia of Data Warehousing and Mining, edited by J. Wang, pp. 1211-1215, copyright 2005 by Information Science Reference, formerly known as Idea Group Reference (an imprint of IGI Global).

Web Usage Mining with Web Logs

Xiangji Huang
York University, Canada

Aijun An,
York University, Canada

Yang Liu
York University, Canada

INTRODUCTION

With the rapid growth of the World Wide Web, the use of automated Web-mining techniques to discover useful and relevant information has become increasingly important. One challenging direction is Web usage mining, wherein one attempts to discover user navigation patterns of Web usage from Web access logs. Properly exploited, the information obtained from Web usage log can assist us to improve the design of a Web site, refine queries for effective Web search, and build personalized search engines.

However, Web log data are usually large in size and extremely detailed, because they are likely to record every aspect of a user request to a Web server. It is thus of great importance to process the raw Web log data in an appropriate way, and identify the target information intelligently. In this chapter, we first briefly review the concept of Web Usage Mining and discuss its difference from classic Knowledge Discovery techniques, and then focus on exploiting Web log sessions, defined as a group of requests made by a single user for a single navigation purpose, in Web usage mining. We also compare some of the state-of-the-art techniques in identifying log sessions from Web servers, and present some popular Web mining techniques, including *Association Rule Mining, Clustering, Classification, Collaborative Filtering, and Sequential Pattern Learning*, that can be exploited on the Web log data for different research and application purposes.

BACKGROUND

Web Usage Mining (WUM), defined as the discovery and analysis of useful information from the World Wide Web, has been an active area of research and commercialization in the recent years (Cooley, Srivastava, & Mobasher, 1997). In general, as shown in Fig1, the WUM process can be considered as a three-phase process, which consists of data preparation, pattern discovery, and pattern analysis (Srivastava, Cooley, Deshpande, & Tan, 2000).

This process implicitly covers the standard process of Knowledge Discovery in the Databases (KDD), and WUM therefore can be regarded as an application of KDD to the Web domain. Nevertheless, it is distinct from standard KDD methods by facing the unique challenge to dealing with the overwhelming resources on the Internet. To assist Web users in browsing the Internet more efficiently, it is widely accepted that the easiest way to find knowledge about user navigations is to explore the Web server logs. Generally, Web logs record all user requests to a Web server. A request is recorded in a log file entry, which contains different types of information, including the IP address of the computer making the request, the user access timestamp, the document or image requested, etc. The following is an example extracted from the *Livelink* Web server log (Huang, An, Cercone & Promhouse, 2002).[1]

Livelink is a database driven web-based knowledge management system developed by Open Text Corporation (http://www.opentext.com). It provides a web-based environment (such as an intranet or extranet) to facilitate collaborations between cross-functional employees within an organization. In this example, a user using the computer with the IP 24.148.27.239 has requested a query with object ID 12856199 on April 10th, at 7:22pm.

However, when statistical methods are applied to such log data, we tend to get results that are too refined or too specific than they should be. In addition, it is very likely that user browsing behaviour is highly uncertain. Different users may visit the same page for

Figure 1. Web usage mining process

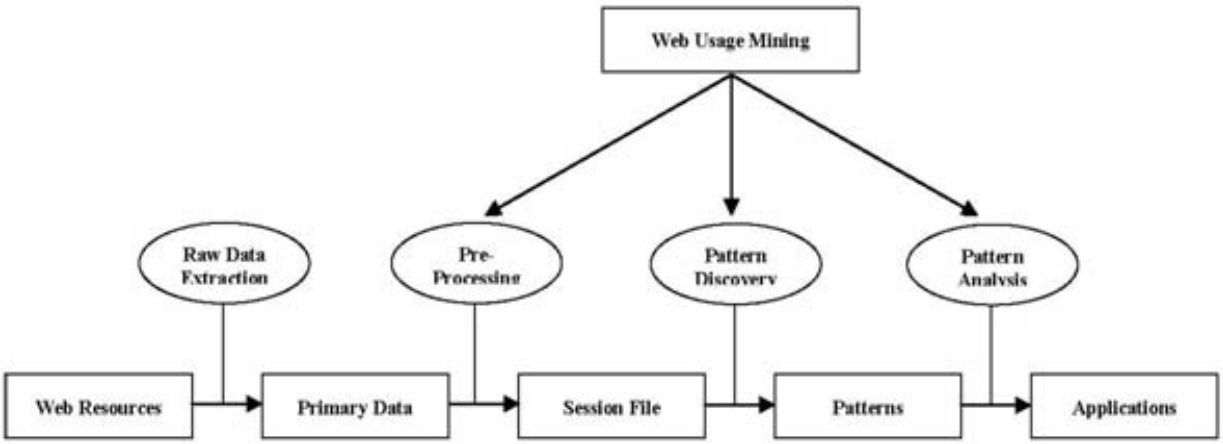


Figura 2. A livelink log entry

```
Wed Apr 10 19:22:52 2002
GATEWAY_INTERFACE = 'CGI/1.1'
HTTPS = 'on'
HTTPS_KEYSIZE = '128'
HTTPS_SECRETKEYSIZE = '1024'
HTTP_ACCEPT_LANGUAGE = 'en-us'
HTTP_CONNECTION = 'Keep-Alive'
HTTP_COOKIE = 'WebEdSessionID=05CAB314874CD61180FE00105A9A1626;
HTTP_HOST = 'intranet.opentext.com'
HTTP_REFERER  ='https://intranet.opentext.com/intranet/livelink.exe?func=doc.
    ViewDoc&nodeId=12856199'
HTTP_USER_AGENT = 'Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)'
objAction = 'viewheader'
objId = '12856199'
QUERY_STRING = 'func=ll&objId=12856199&objAction=viewheader'
REMOTE_HOST = '24.148.27.239'
REQUEST_METHOD = 'GET'
SCRIPT_NAME = '/intranet/livelink.exe'
SERVER_NAME = 'intranet.opentext.com'
SERVER_PORT = '443'
SERVER_PROTOCOL = 'HTTP/1.1'
SERVER_SOFTWARE = 'Microsoft-IIS/5.0'
04/10/2002 19:22:52     Done with Request on socket 069DC4B0
04/10/2002 19:22:57     Processing Request on socket 09A87EF8
```

different purposes, spend various amount of time on the same page, or even access the same page from different sources. Hence, Web usage mining with original entry/ request-based logs may induce erroneous and worthless results. To solve this problem, Web usage session is introduced as a group of requests made by a single user for a single navigation purpose. A user may have a single session or multiple sessions during a period of time, and each session may include one or more requests accomplishing a single task.

MAIN FOCUS OF THE CHAPTER

Session identification method traditionally suffers from the problem of time threshold setting. Different users may have different browsing behaviours, and their

time intervals between sessions may be significantly different. Even for the same user, intervals between sessions may vary. There are several session identification methods reported in the literature.

Timeout

The most commonly used session identification method is called *Timeout*, in which a user session is defined as a sequence of requests from the same user such that no two consecutive requests are separated by an interval more than a predefined threshold. He and Goker (2000) reported the results of experiments that used the timeout method on two different sets of Web logs. By initially selecting a large value, and then gradually decreasing to smaller thresholds, the authors concluded that a time range of 10-15 minutes was an optimal session interval threshold. Catledge and Pitkow (1995) also reported their experimental results where a Web browser was modified to record the time interval between user actions on the browser's interface. One result indicated that the average time interval between two consecutive events by a user was 9.3 minutes, and 25.5 minutes was subsequently recommended as the threshold for session identification. However, the optimal timeout threshold is obviously problem dependent. As an example, Anick (2003) believed that his algorithm favours longer information seeking sessions, and took the time threshold as 60 minutes. Despite the application dependence of the optimal interval length, most commercial products use 30 minutes as a default threshold.

Reference Length

This method assumes that the time a user spends on a page is associated with whether it is an "auxiliary" or "content" page for that user (Cooley, Mobasher & Srivastava, 1999). By analyzing the histogram of page reference lengths, it is found that the time spent on auxiliary pages is usually shorter than that spent on a content page, and the variance of the times spent on auxiliary pages is also smaller. If an assumption is made about the percentage of auxiliary references in a log file, a reference length can be calculated by estimating the optimal cut-off between auxiliary and content references. Once pages are classified as either auxiliary or content pages, a session boundary will be detected whenever a content page is met. However, the potential problem of this method could be that the assumption is valid if only one content page is included in each session. This hypothesis thus may not apply when users visit more than one content page for a single retrieval purpose.

As an extension of the reference length method, a heuristic method, which integrates the reference-oriented method with the timeout strategy, has also been developed (Spilioupoulou, Mobasher, Berendt & Nakagawa, 2003). Nonetheless, the improvement is still limited due to the inherent problems that associate with the two methods.

Maximal Forward Reference

In the method of maximal forward reference, a usage session is defined as the set of pages from the first page in a request sequence to the last page before a backward reference is made (Chen, Park & Yu, 1998). In this approach, a backward reference is naturally defined as a page that has already occurred in the current session. The advantage of this method is that it is not necessary to set any parameters that make assumptions about the characteristics of a particular dataset. Nonetheless, a serious problem could be the backward references will not be recorded on the server if the caching function is enabled at the client side.

Statistical Language Model

Statistical language-model-based approach exploits information theory in the session boundary identification process by measuring the information change in the request sequence (Huang, Peng, An & Schuurmans, 2004). Its objective is to group sequential log entries that are related to a common topic and segment those that are unrelated. They first show that the entropy (or perplexity) of the sequence is always low, when the objects in the sequence are on a common topic and frequently accessed one after another. Also, when a new object that is not relevant to the original topic is observed in the sequence, the new occurrence of this object will induce an increase in entropy, and this new object actually indicates a shift to a new topic. If such changes in entropy pass a threshold, there is reason to claim that the user is currently changing to another topic, and a session boundary can thus be placed before this new object. Through empirical studies, this approach demonstrates favourable results compared with other methods.

Web Mining Methods and Applications of Web Log Data

In order to extract useful user navigation patterns, various data mining techniques, including *Association Rule Mining, clustering, Classification, Collaborative Filtering, and Sequential Pattern Discovery*, have been successfully applied to the pre-processed Web usage session.

1. *Association Rule Mining* - In Web usage mining, association rule mining seeks for a group of objects or pages that are accessed together with a support value exceeding some predefined threshold (Agrawal & Srikant, 1994). For example, it might be interesting to know the correlation between users who view a page containing electronic products and those users who visit the page about sporting equipment in an on-line purchase. As another example, with the assist of association rule mining, page pre-fetching can be applied to reduce the overhead of retrieving documents when loading a page outside memory (Yang, Zhang, & Li, 2001).
2. *Clustering* - Clustering in the Web domain allows us to group together Web users or Web pages or request objects that have similar session characteristics. Current clustering techniques can be classified into two categorizes: discriminative (distance/ similarity-based) approaches and generative (model-based) approaches (Zhong & Ghosh, 2003). Similarity-based clustering determines the distance or (dis) similarity between pair-wise data. As an example, Fu (1999) suggested using URLs to construct a page hierarchy by making use of the directory structure of a web site. The requested sessions of a user was represented with feature vectors by merging access time of visited pages at the same generalized hierarchy level. With the measure of Euclidean distance, the similarities between Web usage sessions were finally calculated for clustering. Model-based clustering places cluster analysis on a principled statistical support. It is based on probability models in which objects are assumed to follow a mixture of probability distributions such that each component distribution stands for a unique cluster (Fraley & Raftery, 2002). The information discovered with clustering algorithms is one of the most important types that have a wide range of applications from real-time personalization to link prediction (Cadez, Heckerman, Meek, Smyth, & White, 2000).
3. *Classification* - Classification is a supervised learning technique that maps Web pages or users into one or more predefined classes. Classification techniques, such as decision tree classifier (Du & Zhan, 2002), naive Bayesian classifier (Geiger & Goldszmidt, 1997), K-nearest-neighbour classifier (Masand, Linoff & Waltz, 1992), neural networks (Ng, Goh, & Low, 1997), language modeling (Mcmahon, Smith 1996), maximum entropy (Nigam, Lafferty & McCallum, 1999), and support vector machines have been widely studied in Web usage mining by taking the advantage of various log data. They can generally be used to categorize visitors or Web pages by selecting features that best describe the properties of their characteristics. For example, a classification rule might be in a format of: 25% of visitors who buy fiction books from Ontario, are aged between 18 and 35, and visit after 5:00pm. Such rules can be applied to categorize future data samples, as well as provide a better understanding of the whole Web data.
4. *Collaborative Filtering* - When a user is navigating on the Web, it is practically infeasible for him/her to find reliable experts that can give advice on which options to choose whenever there is a need. Instead of consulting from a single individual, CF-based methods tend to find opinions of those people having similar interests by measuring the overlaps between their request objects in Web usage sessions (Breese, Heckerman, & Kadie, 1998; Ding & Li, 2005; Kawamae & Takahashi, 2005). To achieve this, the basic idea behind the collaborative filtering technique can be described as follows:
    1. A large group of users' preferences are registered;
    2. With some similarity metric, a subgroup of people that share similar preference of the person who is seeking for advice are selected;
    3. An (weighted) average of the preferences of that subgroup is calculated;

4. A prediction function is applied to recommend options on which the advice seeker has expressed.
   The most commonly used similarity metric is the *Pearson Correlation Coefficients* between the users' preference functions.
5. *Sequential Pattern Mining* - In Web server logs, user click streams are sequentially recorded. Each Web usage session is a sequence of pages or objects that a user requests during a visit to the Web site. The discovery of sequential patterns in Web server logs allows for predicting user navigation patterns (Gunduz& Ozsu, 2003; Mobasher, Dai, Luo, & Nakagawa, 2002). By analyzing the sequential patterns, the Web mining system can determine temporal relationships among data items.

FUTURE TRENDS

Some questions involving both session boundary identification and Web usage mining applications are still open for further investigations. For example, although the language-model-based session identification approach demonstrates the best performance compared with other methods, it is still sensitive to the setting of entropy threshold. An inappropriate threshold value may yield non-optimal results. An automatic parameter estimation method therefore should be investigated. Some future trends also include combining Web usage mining approaches with the Web content learning. For example, the information gained through discovering the Web user/page patterns can be used to refine queries of a personalized search engine, or construct an adaptive information system.

CONCLUSION

To better satisfy the requirements of various Web-based applications, Web usage mining exploits data mining methods to discover useful patterns from Web log data. In this chapter, we first reviewed the concept of Web usage mining, and discussed its difference from standard KDD methods. To make a better understanding of the unique characteristics in Web data, we then explained the need of introducing Web usage sessions, and focused on describing different state-of-the-art session identification techniques. We finally presented some popular Web usage mining techniques, and displayed their applications in recent research.

REFERENCES

Agrawal, R., & Srikant, R. (1994). Fast Algorithms for Mining Association Rules in Large Databases. *International Conference on Very Large Data Bases*, 487-499.

Anick, P. (2003). Using Terminological Feedback for Web Search Refinement: A Log-Based Study, *SIGIR*, 88-95.

Breese, J., Heckerman, D. & Kadie, C. (1998). Empirical Analysis of Predictive Algorithms for Collaborative Filtering, *the 14th Uncertainty in Artificial Intelligence*, 43-52.

Cadez, I., Heckerman, D., Meek, C., Smyth, P. & White, S (2000). Visualization of Navigation Patterns on a Web Site Using Model-Based Clustering. *KDD-2000*.

Catledge, L.D. & Pitkow, J. (1995). Characterizing Browsing Strategies in the World-Wide Web, *Computer Networks ISDN System*, 27:1065-1073.

Chen, M., Park, J. S., & Yu, P. S. (1998). Efficient Data Mining for Path Traversal Patterns. *IEEE Trans. Knowl. Data Eng*, 10(2): 209-221.

Cooley, R., Mobasher, B., & Srivastava, J. (1999). Data Preparation for Mining World Wide Web Browsing Patterns, *Knowledge Information System*, 1(1): 5-32.

Cooley, R., Srivastave, J., & Mobasher, B. (1997). Web Mining: Information and Pattern Discovery on the World Wide Web, *the 9th IEEE International Conference on Tools with Artificial Intelligence.*

Ding, Y. & Li, X. (2005). Time Weight Collaborative Filtering, *14th Conference on Information and Knowledge Management*, 485-492.

Du, W. and Zhan, Z. (2002). Building Decision Tree Classifier on Private Data. *IEEE ICDM Workshop on Privacy, Security and Data Mining*, 1-8.

Fraley, C., & Raftery, A.E. (2002). Model-based Clustering, Discriminant Analysis and Density Estimation, *Journal of the American Statistical Association*, 611- 631.

Fu, Y., Sandhu, K., & Shih, M. (1999). Generalization-based Approach to Clustering of Web Usage Sessions, *WEBKDD*, 21–38.

Geiger, D. & Goldszmidt, M. (1997). Bayesian Networks Classifiers. *Machine Learning*, 29:131-163.

Gunduz, S. & Ozsu, M. T. (2003). A Web Page Prediction Model based on Click-stream Tree Representation of User Behavior, *the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 535-540.

He, D., & Goker, A. (2000). Detecting Session Boundaries from Web User Logs, *the BCS-IRSG 22nd Annual Colloquium on Information Retrieval Research*.

Huang, X., Peng, F., An, A., & Schuurmans, D. (2004). Dynamic Web Log Session Identification with Statistical Language Models, *Journal of the American Society for Information Science and Technology (JASIST)*, 55(14), 1290-1303.

Huang, X., An, A., Cercone, N. and Promhouse, G. (2002). Discovery of Interesting Association Rules from Livelink Web Log Data, *Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM'02)*, Maebashi City, Japan, December 9-12, 2002, 763-766.

Liu, Y., Huang, X. & An, A. (2007). Personalized Recommendation with Adaptive Mixture of Markov Models, *Journal of the American Society for Information Science and Technology (JASIST)*, 58(12), 1851-1870.

Liu, Y., Huang, X., An, A. & Yu, X. (2007). ARSA: A Sentiment-Aware Model for Predicting Sales Performance Using Blogs, *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'07)*, Amsterdam, July 23-27, 2007.

Kawamae, N., & Takahashi, K. (2005). Information Retrieval based on Collaborative Filtering with Latent Interest Semantic Map. *Conference on Knowledge Discovery in Data*, 618-623.

Masand, B., Linoff, G., & Waltz, D. (1992). Classifying News Stories Using Memory Based Reasoning. *The 15th Annual ACM/SIGIR Conference*, 59-65.

Mcmahon JG, & Smith FJ. (1996). Improving Statistical Language Model Performance with Automatically Generated Word Hierarchies. *Comput. Linguist.*, 22(2):217-247.

Mobasher, B., Dai, H., Luo, T., & Nakagawa, M. (2002). Using Sequential and Non-Sequential Patterns in Predictive Web Usage Mining Tasks. Proceedings of the 2002 *IEEE International Conference on Data Mining (ICDM'02)*, Maebashi City, Japan, December 9-12, 2002, 669-672.

Ng, H., Goh, W., & Low, K. (1997). Feature Selection, Perceptron Learning, and a Usability Case Study for Text Categorization. *ACM/SIGIR Conference*, 67-73.

Nigam, K., Lafferty, J., & McCallum, A. (1999). Using. Maximum Entropy for Text Classification. *IJCAI-99 Workshop on Machine Learning for Information Filtering*, 61–67.

Srivastava, J., Cooley, R., Deshpande, M., & Tan, P. (2000). Web Usage Mining Discovery and Applications of Usage Patterns form Web Data, *SIGKDD Explorations, 1(2)*.

Spilioupoulou, M., Mobasher, B., Berendt, B., & Nakagawa M. (2003). A Framework for the Evaluation of Session Reconstruction Heuristics in Web Usage Analysis, *INFORMS Journal of Compting*, 15(2), 1-34.

Yang, Q., Zhang, H. H., & Li, T. Y. (2001). Mining Web Logs for Prediction Models in WWW Caching and Prefetching, *the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 473-478.

Zhong, S. & Ghosh, J. (2003). A Unified Framework for Model-based Clustering, *Journal of Machine Learning Research*, 1001-1037.

ENDNOTE

[1] For security reasons, some of the lines are removed.

KEY TERMS

Session Identification: The application of determining the boundaries of Web usage sessions.

Statistic Language Model: The probability distributions defined on sequences of words, $P(w_1 \ldots w_n)$. In

speech recognition, such models refer to a probabilistic distribution capturing the statistics of the generation of a language, and attempt to predict the next word in a speech sequence.

Web Log Mining: The application of discovering Web usage patterns from the records of Web server logs.

Web Usage Mining: The application of data mining techniques to discover usage patterns from Web data.

Web Usage Session: A group of requests made by a single user for a single navigation purpose.

Wrapper Feature Selection

Kyriacos Chrysostomou
Brunel University, UK

Manwai Lee
Brunel University, UK

Sherry Y. Chen
Brunel University, UK

Xiaohui Liu
Brunel University, UK

INTRODUCTION

It is well known that the performance of most data mining algorithms can be deteriorated by features that do not add any value to learning tasks. Feature selection can be used to limit the effects of such features by seeking only the relevant subset from the original features (de Souza et al., 2006). This subset of the relevant features is discovered by removing those that are considered as irrelevant or redundant. By reducing the number of features in this way, the time taken to perform classification is significantly reduced; the reduced dataset is easier to handle as fewer training instances are needed (because fewer features are present), subsequently resulting in simpler classifiers which are often more accurate.

Due to the abovementioned benefits, feature selection has been widely applied to reduce the number of features in many data mining applications where data have hundreds or even thousands of features. A large number of approaches exist for performing feature selection including filters (Kira & Rendell, 1992), wrappers (Kohavi & John, 1997), and embedded methods (Quinlan, 1993). Among these approaches, the wrapper appears to be the most popularly used approach. Wrappers have proven popular in many research areas, including Bioinformatics (Ni & Liu, 2004), image classification (Puig & Garcia, 2006) and web page classification (Piramuthu, 2003). One of the reasons for the popularity of wrappers is that they make use of a classifier to help in the selection of the most relevant feature subset (John et al., 1994). On the other hand, the remaining methods, especially filters, evaluate the merit of a feature subset based on the characteristics of the data and statistical measures, e.g., chi-square, rather than the classifiers intended for use (Huang et al., 2007). Discarding the classifier when performing feature selection can subsequently result in poor classification performance. This is because the relevant feature subset will not reflect the classifier's specific characteristics. In this way, the resulting subset may not contain those features that are most relevant to the classifier and learning task. The wrapper is therefore superior to other feature selection methods like filters since it finds feature subsets that are more suited to the data mining problem.

These differences between wrappers and other existing feature selection techniques have been reviewed by a number of studies (e.g. Huan & Lei, 2005). However, such studies have primarily focussed on providing a holistic view of all the different types of techniques. Although these works provide comprehensive information, there has yet to appear a deep review that solely focuses on the most popular feature selection technique; the wrapper. To this end, this paper aims to present an in-depth survey of the wrapper. In particular, attention will be given to improvements made to the wrapper since they are known for being much slower than other existing feature selection techniques. This is primarily because wrappers are required to repeatedly run the classifier when determining feature accuracy and perform feature selection again each time a different classifier is used. To overcome such problems, researchers in this area have spent considerable effort in improving the performance of wrappers (Yu & Cho, 2006). Basically, improvements can be divided into two trends. One focuses on reducing the time taken to do feature selection and the other emphasises on improv-

ing the accuracy of the selected subset of features. In fact, there are close relationships between these two trends because one can potentially influence the other. In other words, decreasing the time taken to perform feature selection with wrappers may potentially affect the accuracy of the final output. This relationship has been investigated by several studies and will be reviewed in this paper.

The paper will first formally define the feature selection process, with an emphasis on the wrapper approach. Subsequently, improvements made to the wrapper for reducing the time taken to do feature selection and increasing the overall accuracy of the selected subset of features will be discussed. It then moves to discuss future directions for wrapper feature selection approaches. Finally, conclusions are drawn at the end of the paper.

BACKGROUND

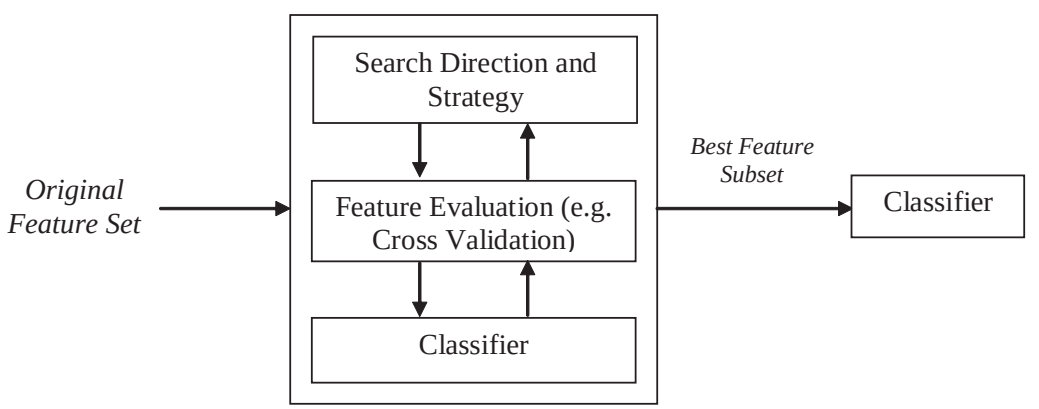
Typically, feature selection can be formally defined in the following manner. Suppose F is the given set of original features with cardinality n (where n symbolises the number of features in set F), and $\overline{F}$ is the selected feature subset with cardinality $\overline{n}$ (where $\overline{n}$ symbolises the number of features in set $\overline{F}$), then $\overline{F} \subseteq F$. Also, let $J(\overline{F})$ be the selection criterion for selecting feature set $\overline{F}$. We assume that a higher value of J indicates a better feature subset. Thus, the goal is to maximise $J()$. The problem of feature selection is to find a subset of features $\overline{F} \subseteq F$ such that,

$$J(\overline{F}) = \max_{Z \subseteq F, |Z|=n} J(Z)$$

Deriving a feature subset that maximises $J(\ )$ typically consists of four key steps namely, search direction, search strategy, subset evaluation and stopping criterion (Huan & Lei, 2005). Search direction defines the point at which the search for the most relevant subset will begin. Complimentary to the direction of the search is the search strategy. The search strategy outlines the way in which feature subsets are searched within the feature space. Each of the feature subsets found is then evaluated according to some evaluation criteria. Finally, a stopping criterion is used for halting the search through feature subsets.

As indicated in the Introduction, this paper will focus on the wrapper approach, where a classifier is used as the evaluation criterion for maximising $J(\ )$. Basically, the wrapper uses the classifier as a black box. The classifier is repeatedly run on the dataset using various subsets of the original features. These feature subsets are found through the use of a search strategy. The classifier's performance and some accuracy estimation method like cross validation are then used to evaluate the accuracy of each subset (John et al., 1994). The feature subset with the highest accuracy is chosen as the final set on which to run the classifier.

Figure 1. Wrapper feature selection



To have a better understanding of the wrapper approach, this paper will use a decision tree as an example to explain the aforementioned process. The decision tree classifier initially seeks and generates different feature subsets with some search strategies, one of which is forward selection. The forward selection strategy begins with no features and successively adds more features that are deemed relevant by the decision tree classifier. The feature subsets found using this strategy may then be evaluated using the decision tree's performance and 10-fold cross validation, where the subset with the highest accuracy is determined as the most relevant. This relevant subset is subsequently fed as input to the decision tree classifier. The entire procedure for performing wrapper feature selection is illustrated in Figure 1.

DECREASE IN TIME

Although the wrapper is able to find accurate feature subsets, the main criticism of the approach is the amount of time needed to perform feature selection. Typically, the wrapper will require considerably more time to examine feature subsets compared to any other feature selection approaches. This is because of two issues. The first issue concerns the use of a classifier. By using a classifier, more time is needed in examining each potential feature subset searched in the feature space. The second issue regards the use of cross validation. Cross validation is used in conjunction with the classifier to determine the level of accuracy of feature subsets. When both the classifier and cross validation are used together, the wrapper runs prohibitively slow. These drawbacks have led researchers to investigate ways of reducing the time of the wrapper process.

As stated above, the classifier is one of the main reasons why the wrapper performs slower than other feature selection approaches. To alleviate the effects of using a classifier, Caruana & Freitag (1994) developed a new method for speeding up the wrapper approach when specifically used with decision tree classifiers. The method functions by reducing the number of decision trees grown during feature selection by recording the features that were used to construct the trees. By doing so, less time is needed to analyze the features used in tree formation. In addition to two well known decision tree classifiers, ID3 and C4.5, five different search strategies were used to test the effectiveness of the method, including forward selection, backward elimination, forward stepwise selection, backward stepwise elimination, and backward stepwise elimination-SLASH, which is a bi-directional version of backward stepwise elimination. Empirical analysis revealed that, irrespective of the search strategy and decision tree classifier used, the time taken to perform feature selection decreased.

Furthermore, Kohavi & Sommerfield (1995) introduced the concept of 'compound' operators in an attempt to make the wrapper perform in less time. The purpose of using compound operators is to direct the search strategy more quickly toward the most relevant features. In this way, the classifier will need to spend less time evaluating all the features. Experiments using the compound operators were carried out using two classifiers: ID3 and Naïve Bayes. Results showed a significant decrease in the amount of time needed to perform feature selection when either classifier was used. Improvements in classification accuracy for ID3 and Naïve Bayes were also found when compound operators were implemented.

Besides the classifiers, cross validation is another factor that can decrease the speed at which the wrapper performs features selection. A strategy for reducing the wrapper's time complexity when used in conjunction with cross validation was presented by Moore & Lee (1994). The strategy reduces the number of instances used during the evaluation stage of feature selection so the cost of fully evaluating each feature subset is also decreased. They showed that the new strategy successfully reduced the number of feature subsets evaluated during feature selection. This led to a drop in the amount of time needed to perform wrapper feature selection. It was also found that the reliability of the chosen feature subset was unaffected by the fall in the number of instances.

Hashemi (2005) also investigated the effects of reducing the number of instances used for feature selection. Hashemi presented a new wrapper approach that performs feature selection roughly 75 times faster than traditional wrapper approaches. The new wrapper approach does this by using an algorithm called Atypical Sequential Removing (ASR). The ASR algorithm finds and removes those instances in the data, which do not influence classifier performance. By decreasing the number of instances, the process of feature selection can be speeded up as there will be less data to deal with. Experiments were carried out using the proposed

wrapper approach with different classifiers, including Support Vector Machines (SVM), *k*-Nearest Neighbour and C4.5. Overall, findings showed that although the accuracy of some classifiers did not improve when compared to the use of all instances, the new wrapper method performed much faster.

INCREASE OF ACCURACY

The aforementioned studies have so far shown that it is possible to reduce the time needed to perform wrapper feature selection. Although time complexity is a major issue, especially when many features are involved, the accuracy of the chosen feature subset is also very important. The idea of accuracy and time is interrelated because improving one may affect the other. Numerous studies have investigated this relationship by using evolutionary search strategies called Genetic Algorithms (GA) (e.g. Ni & Liu, 2004) because GA possess powerful search capabilities (Sikora & Piramuthu, 2007). Rhitoff, et al. (2002) is an example of works using GA. They incorporate GA with the wrapper to form a feature selection technique that avoids a suboptimal solution without sacrificing much in speed. Specifically, the GA wrapper uses a SVM as the classifier when performing feature selection. Results showed that accuracy significantly improved when compared to using no feature selection. Their approach was also tested against the well known sequential forward selection wrapper with similar findings.

Another framework combining the uses of GA and feature selection approaches can be found in Sikora & Piramuthu (2007). This framework uses GA with the Hausdorff distance measure for wrapper feature selection. Experimental results comparing this framework to a GA-based wrapper approach without the Hausdorff distance measure showed that it provided superior performance. The GA and Hausdorff wrapper feature selection approach was not only able to improve classification accuracy by 10% but also reduce the amount of time by 60%. This goes to show that the accuracy of the wrapper approach can be improved no matter if time taken is lowered.

Furthermore, Ruiz et al. (2006) developed a new gene selection method called Best Incremental Ranked Subset (BIRS) based on the wrapper approach. The method aims to improve classification accuracy of cancer data without affecting the time taken to do feature selection. BIRS does this by first ranking the genes. A small subset of genes with the highest rank is then fed as input to the wrapper. The method was tested using three different classifiers, i.e., Naïve Bayes, Nearest Neighbour, and C4.5, on four DNA microarray datasets. Experimental results on these datasets showed that BIRS is a very fast feature selection approach when compared to a wrapper that uses all the genes. In addition, BIRS was found to produce good classification accuracy.

The abovementioned studies have clearly shown that the wrapper's time complexity can significantly be reduced. In addition, we have seen that the accuracy of the wrapper can be improved. However, improving time and accuracy only present the first step towards developing more robust wrapper approaches. Further research is needed to identify alternative ways of improving the way in which the wrapper performs feature selection. Other directions for further research are presented in the next section.

FUTURE TRENDS

The reviewed studies demonstrate that feature selection, more specifically the wrapper, is a very popular data analysis tool. However, there are some limitations that need to be further investigated.

Combined Feature Selection

As described in the previous section, the classifier is used by the wrapper to select the most relevant feature subset. In this way, the biases specific to the classifier are included in the feature selection process, which often lead to subsets that are better suited to the classifier. However, each classifier has its own limitations and current research mainly uses a single classifier when selecting the most relevant features. There exists little work that considers combining several different classifiers and deriving a consensus from the features selected from these different classifiers. Such a combined approach can be used to overcome the limitations of each individual classifier.

Search Techniques

The wrapper approach is not only influenced by the classifier used, but also by the strategy applied to

search through the vast number of feature subsets. Often, selecting the right strategy can be difficult as there are many to choose from. In addition, wrongly chosen search strategies can influence the accuracy of the output feature subset. To alleviate the task of choosing a search strategy, it would be interesting to investigate search strategies that use concepts from Soft Computing, such as Fuzzy Logic and Rough Set Theory. This is because Soft Computing methods are very good at handling ill-defined learning tasks. However, very little work has been done to implement these strategies in feature selection, so more empirical studies are needed to identify their true usefulness and effectiveness.

Parallelization of Feature Selection

In recent years, the notion of parallelization for improving the performance of data mining techniques has received much interest from researchers. Such parallelization is conducted by running data mining techniques over several computers to output results in less time. However, not many attempts have been made to apply parallelization to feature selection (de Souza et al., 2006). Feature selection approaches, specifically the wrapper, could benefit from such an approach. More specifically, parallelization can reduce the amount of time needed to perform feature selection by distributing the workload among different computers and their processors. In this way, much larger datasets can be analyzed in much less time. Due to such advantages, further investigation is certainly needed to evaluate the effectiveness of applying parallelization for the wrapper or any other feature selection approach.

CONCLUSION

In summary, feature selection is a very useful approach for choosing a small subset of features that is most relevant to a learning task. Reducing the number of features in this way leads to many benefits, including more concise results with better comprehensibility. The most popular feature selection approach used to reduce the number of features is the wrapper. In this paper, we provided a review of the wrapper feature selection process. Particular attention was given to survey current works that reduce the time complexity and increase the accuracy of the wrapper. However, these works apply the wrapper, and its improved variants, to solve different data mining problems. Since each problem will be different in nature, it would be beneficial to collectively integrate these works into a generic guide to help researchers decide the most suitable way for applying the wrapper to a particular data mining problem. By doing so, researchers can exploit the full potential of the wrapper approach to gain a relevant feature subset that is more accurate and suitable.

ACKNOWLEDGMENT

The work presented in this paper is funded by the UK Arts and Humanities Research Council (reference: MRG/AN9183/APN16300).

REFERENCES

Caruana, R. & Freitag, D. (1994). Greedy attribute selection. In *Machine Learning: Proceedings of the Eleventh International Conference*, Morgan Kaufmann.

de Souza, J.T., et al. (2006). Parallelizing Feature Selection. *Algorithmica*, 45(3), 433-456.

Hashemi, S. (2005). Linear-time wrappers to identify atypical points: two subset generation methods. *IEEE Transactions on Knowledge and Data Engineering*, 17(9), 1289-1297.

Huan, L. & Lei, Y. (2005). Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering*, 17(4), 491-502.

Huang, J., et al. (2007). An intelligent learning diagnosis system for Web-based thematic learning platform. *Computers & Education*, 48(4), 658-679.

John, G.H., et al. (1994). Irrelevant features and the subset selection problem. *In Proceedings of the Eleventh International Conference on Machine learning*, pp 121–129, New Brunswick, NJ, 1994. Morgan Kaufmann.

Kira, K. & Rendell, L.A. (1992). A practical approach to feature selection. In *Machine Learning: Proceedings of the Ninth International Conference*, pp 249-256.

Kohavi, R. & John, G.H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2), 273-324.

Kohavi, R. & Sommerfield, D. (1995). Feature subset selection using the wrapper method: Overfitting and dynamic search space topology. In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, pp 192-197 AAAI Press.

Moore, A.W. & Lee, M.S. (1994). Efficient algorithms for minimizing cross validation error. In *Machine Learning: Proceedings of the Eleventh International Conference*, pp 190-198, Morgan Kaufmann.

Ni, B. & Liu, J. (2004). A hybrid filter/wrapper gene selection method for microarray classification. *Proceedings of 2004 International Conference on Machine Learning and Cybernetics*, 4, 2537-2542.

Piramuthu, S. (2003). On learning to predict Web traffic. *Decision Support Systems*, 35(2), 213-229.

Puig, D. & Garcia, M.A. (2006). Automatic texture feature selection for image pixel classification. *Pattern Recognition*, 39(11), 1996-2009.

Quinlan, J.R. (1993). *C4.5: Programs for Machine Learning*, San Francisco, CA: Morgan Kaufmann.

Ritthoff, O., et al. (2002). A Hybrid Approach to Feature Selection and Generation Using an Evolutionary Algorithm. In *Proceedings of the UKCI-02*, Birmingham, UK, pp. 147-154.

Ruiz, R., et al. (2006). Incremental wrapper-based gene selection from microarray data for cancer classification. *Pattern Recognition*, 39(12), 2383-2392.

Sikora, R. & Piramuthu, S. (2007). Framework for efficient feature selection in genetic algorithm based data mining. *European Journal of Operational Research*, 80(2), 723-737.

Yu, E. & Cho, S. (2006). Ensemble based on GA wrapper feature selection, *Computers &Industrial Engineering*, In Press.

KEY TERMS

Classification: Prediction of the class label of a new instance using a set of already labelled data instances.

Classifier: Categorises instances based on a labelled dataset and outputs a classification model.

Cross Validation: An accuracy estimation technique typically used for classification and feature selection, which estimates generalisation error and prevents overfitting of the data. Cross validation is extremely useful when a separate validation or test set is unavailable.

Decision Tree: A top-down hierarchical structure constructed by repeatedly splitting the entire dataset into two or more nodes according to some target variable.

Genetic Algorithm: A powerful search technique that uses concepts from biological evolution. It is initiated with a population of individual solutions called chromosomes. Genetic operators such as crossover and mutation are then used to select the best solution.

***k*-Nearest Neighbour:** A technique that uses a distance metric to select the *k* closest instance(s) to a new instance. The class label of the closest instance will be assigned to the new instance. This technique can be applied to both classification and regression problems.

Support Vector Machine (SVM): The mapping of the original feature space onto a higher dimensional space to find an optimal hyperplane that correctly classifies instances in a dataset by maximising the margin between instances and minimising the chance of errors.

XML Warehousing and OLAP

Hadj Mahboubi
University of Lyon (ERIC Lyon 2), France

Marouane Hachicha
University of Lyon (ERIC Lyon 2), France

Jérôme Darmont
University of Lyon (ERIC Lyon 2), France

INTRODUCTION

With the eXtensible Markup Language (XML) becoming a standard for representing business data (Beyer et al., 2005), a new trend toward XML data warehousing has been emerging for a couple of years, as well as efforts for extending the XQuery language with near On-Line Analytical Processing (OLAP) capabilities (grouping, aggregation, etc.). Though this is not an easy task, these new approaches, techniques and architectures aim at taking specificities of XML into account (e.g., heterogeneous number and order of dimensions or complex measures in facts, ragged dimension hierarchies…) that would be intricate to handle in a relational environment.

The aim of this article is to present an overview of the major XML warehousing approaches from the literature, as well as the existing approaches for performing OLAP analyses over XML data (which is termed XML-OLAP or XOLAP; Wang et al., 2005). We also discuss the issues and future trends in this area and illustrate this topic by presenting the design of a unified, XML data warehouse architecture and a set of XOLAP operators expressed in an XML algebra.

BACKGROUND

XML warehousing research may be subdivided into three families. The first family focuses on Web data integration for decision-support purposes. However, actual XML warehouse models are not very elaborate. The second family of approaches is explicitly based on classical warehouse logical models (star-like schemas). The third family we identify relates to document warehousing. In addition, recent efforts aim at performing OLAP analyses over XML data.

XML Web Warehouses

The objective of these approaches is to gather XML Web sources and integrate them into a data warehouse. For instance, Xyleme (2001) is a dynamic warehouse for XML data from the Web that supports query evaluation, change control and data integration. No particular warehouse model is proposed, though.

Golfarelli et al. (2001) propose a semi-automatic approach for building a data mart's conceptual schema from XML sources. The authors show how multidimensional design may be carried out starting directly from XML sources and propose an algorithm for correctly inferring the information needed for data warehousing.

Finally, Vrdoljak et al. (2003) introduce the design of a Web warehouse that originates from XML Schemas describing operational sources. This method consists in preprocessing XML Schemas, in creating and transforming the schema graph, in selecting facts and in creating a logical schema that validates a data warehouse.

XML Data Warehouses

In his XML-star schema, Pokorný (2002) models a star schema in XML by defining dimension hierarchies as sets of logically connected collections of XML data, and facts as XML data elements.

Hümmer et al. (2003) propose a family of templates enabling the description of a multidimensional structure for integrating several data warehouses into a virtual

or federated warehouse. These templates, collectively named XCube, consist of three kinds of XML documents with respect to specific schemas: XCubeSchema stores metadata; XCubeDimension describes dimensions and their hierarchy levels; and XCubeFact stores facts, i.e., measures and the corresponding dimensions.

Rusu et al. (2005) propose a methodology, based on the XQuery technology, for building XML data warehouses, which covers processes such as data cleaning, summarization, intermediating XML documents, updating/linking existing documents and creating fact tables. Facts and dimensions are represented by XML documents built with XQueries.

Park et al. (2005) introduce an XML warehousing framework where every fact and dimension is stored as an XML document. The proposed model features a single repository of XML documents for facts and multiple repositories for dimensions (one per dimension).

Eventually, Boussaïd et al. (2006) propose an XML-based methodology, X-Warehousing, for warehousing complex data (Darmont et al., 2005). They use XML Schema as a modeling language to represent users' analysis needs, which are compared to complex data stored in heterogeneous XML sources. Information needed for building an XML cube is then extracted from these sources.

XML DOCUMENT WAREHOUSES

Baril and Bellahsène (2003) envisage XML data warehouses as collections of materialized views represented by XML documents. Views allow filtering and restructuring XML sources, and provide a mediated schema that constitutes a uniform interface for querying the XML data warehouse. Following this approach, the authors have developed the DAWAX system.

Nassis et al. (2005) propose a conceptual approach for designing and building an XML repository, named xFACT. They exploit object-oriented concepts and propose to select dimensions based on user requirements. To enhance the XML data warehouse's expressiveness, these dimensions are represented by XML virtual views. In this approach, the authors assume that all dimensions are part of fact data and that each fact is described in a single XML document.

Rajugan et al. (2005) also propose a view-driven approach for modeling and designing an XML fact repository, named GxFact. GxFact gathers xFACTs (distributed XML warehouses and datamarts) in a global company setting. The authors also provide three design strategies for building and managing GxFact to model further hierarchical dimensions and/or global document warehouses.

Finally, Zhang et al. (2005) propose an approach to materialize XML data warehouses based on frequent query patterns discovered from historical queries. The authors apply a hierarchical clustering technique to merge queries and therefore build the warehouse.

OLAP ANALYSES OVER XML DATA

Chronologically, the first proposals for performing OLAP analyses over XML data mainly rely on the power of relational implementations of OLAP, while more recent research directly relates to XOLAP.

Jensen et al. (2001) propose an integration architecture and a multidimensional UML model for relational and XML data. They also discuss the design of XML databases supporting OLAP analyses. The output of this process is a unified relational representation of data, which are queried with the OQL language. Niemi et al. (2002) follow the same logic, but propose a dedicated language named MDX. In both these approaches, XML data are mapped into a relational database and exploited with relational query languages. Hence, no XML-specific OLAP operator is defined.

Pedersen et al. (2004) also advocate for federating XML data and existing OLAP cubes, but in addition, they propose an algebra composed of three operators. The most fundamental operator in an OLAP-XML federation, decoration, attaches a new dimension to a cube with respect to linked XML elements. Selection and generalized projection help filter and aggregate fact measures, respectively. They more or less correspond to the classical OLAP slice and dice operators. These three operators are implemented with an extension of the SQL_M language, SQL_{XM}, which helps associate XPath queries to SQL_M queries. SQL_M is itself an extension of SQL for processing multidimensional data.

Eventually, Park et al. (2005) propose an OLAP framework for XML documents called XML-OLAP and introduce the notion of XML cube (XQ-Cube). A specific multidimensional language (XML-MDX) is applied on XQ-Cubes. Wang et al. (2005) also propose a general aggregation operator for XML, GXaggrega-

tion, which forms the base of XCube, an extension of the traditional cube operator for XML data. These two operators have been implemented as an extension of XQuery. In opposition, Wiwatwattana et al. (2007) argue that such an extension from the relational model cannot address the specific issues posed by the flexibility of XML. Hence, they propose an XML warehouse-specific cube lattice definition, a cube operator named X^3, and a generalized specification mechanism. They also discuss the issue of cube computation and compare several alternative algorithms. X^3 has been implemented in C++ within the TIMBER XML-native Database Management System (DBMS).

UNIFIED XML WAREHOUSING AND XOLAP MODELS

XML Warehouse Architecture

Previous XML warehousing approaches assume that the warehouse is composed of XML documents that represent both facts and dimensions. All these studies more or less converge toward a unified XML warehouse model. They mostly differ in the way dimensions are handled and the number of XML documents that are used to store facts and dimensions.

A performance evaluation study of these different representations has been performed by Boukraa et al. (2006). It showed that representing facts in one single XML document and each dimension in one XML document allowed the best performance.

Moreover, this representation also allows to model constellation schemas without duplicating dimension information. Several fact documents can indeed share the same dimensions. Also, since each dimension and its hierarchy levels are stored in one XML document, dimension updates are more easily and efficiently performed than if dimensions were either embedded with the facts or all stored in one single document.

Hence, we propose to adopt this architecture model to represent XML data warehouses. It is actually the translation of a classical snowflake schema into XML. More precisely, our reference data warehouse is composed of the following XML documents:

1. *dw-model.xml* represents the warehouse metadata (basically the warehouse schema);
2. *facts.xml* helps store the facts, i.e., dimension identifiers and measure values;
3. *dimension$_d$.xml* helps store a given dimension *d*'s attribute values.

Figure 1 represents the *dw-model.xml* document's graph structure. *dw-model.xml* defines the multidimensional structure of the warehouse. Its root node, *DW-model*, is composed of two types of nodes: *dimension* and *FactDoc*. A *dimension* node defines one dimension, its possible hierarchical levels (*Level* elements) and attributes (including their types), as well as the

Figure 1. Dw-model.xml graph structure

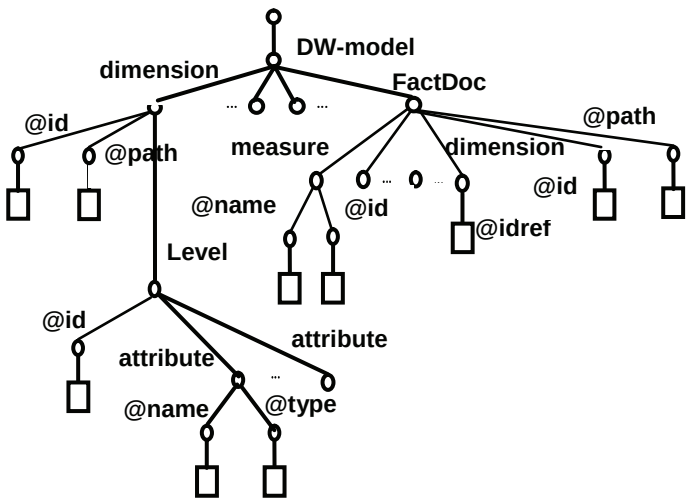


Figure 2. Facts.xml graph structure

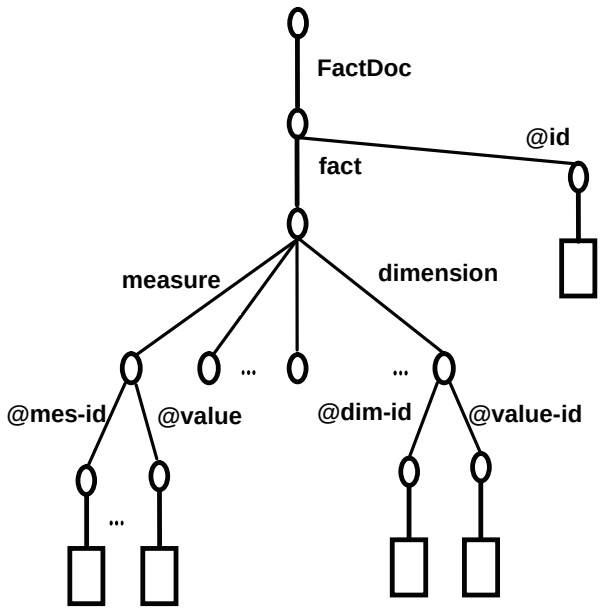
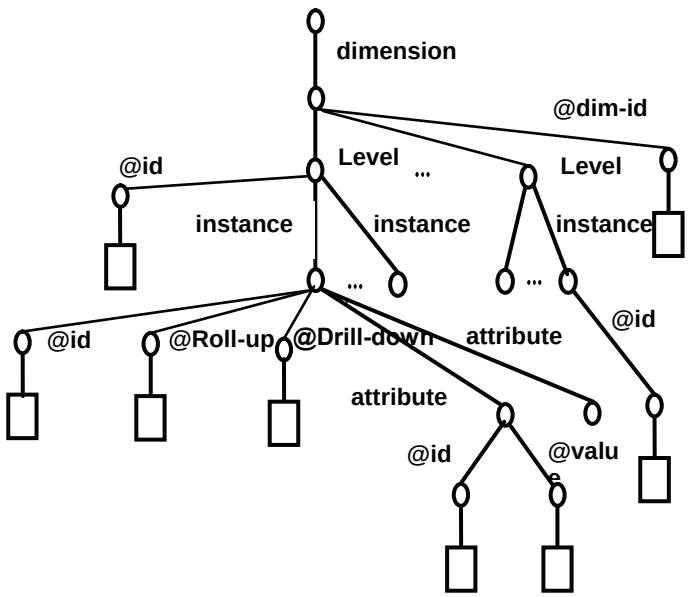


Figure 3. Dimension$_d$.xml graph structure



path to the corresponding *dimension$_d$.xml* document. A *FactDoc* element defines a fact, i.e., its measures, internal references to the corresponding dimensions, and the path to the *facts.xml* document. Note that several *FactDoc* elements are possible, thus enabling constellation schemas.

Figure 2 represents the *facts.xml* document's graph structure. *facts.xml* stores the facts and is composed of *fact* nodes defining measures and dimension references. The document root node, *FactDoc*, is composed of *fact* sub-elements, each of whose instantiates a fact, i.e., measure values and dimension references. These

identifier-based references support the fact-to-dimension relationship.

Finally, Figure 3 represents the *dimension$_d$.xml* document's graph structure. *dimension$_d$.xml* helps instantiate dimension *d,* including any hierarchy level. Its root node, *dimension,* is composed of *Level* nodes. Each one defines a hierarchy level composed of *instance* nodes that each defines the level's attribute values. In addition, an *instance* element contains *Roll-up* and *Drill-down* attributes that define the hierarchical relationship within dimension *d*.

Algebraic Expression of XOLAP Operators

These last decade's efforts for formalizing OLAP algebras (Agrawal et al., 1997; Gyssens and Lakshmanan, 1997; Thomas and Datta, 2001; Ravat et al., 2006) have helped design a formal framework and well-identified operators. Existing OLAP operators being defined in either a relational or multidimensional context, they must now be adapted to the data model of XML documents (namely trees or, more generally, graphs) and enriched with XML-specific operators.

Existing approaches that aim at XOLAP do not fully satisfy these objectives. Some favor the translation of XML data cubes in relational, and query them with extensions of the SQL language. Others tend toward multidimensional solutions that exploit XML query languages such as XQuery or XML-MDX. However, in terms of algebra, these works only propose a fairly limited number of operators.

As Wiwatwattana et al. (2007), we aim at a native-XML solution that exploits XQuery. As a first step toward an XOLAP platform, we initiated a previously inexistent formal framework in the XML context by demonstrating how the TAX Tree Algebra for XML (Jagadish et al., 2001) could support OLAP operators. Among the many XML algebras from the literature, we selected TAX for its richness. TAX indeed includes, under its logical and physical forms, more than twenty operators, which allows us many combinations for expressing XOLAP operators. Furthermore, TAX's expressivity is widely acknowledged, since this algebra can be expressed with most XML query languages, and especially XQuery, which we particularly target because of its standard status. Finally, TAX and its derived algebra TLC (Paparizos et al., 2004) provide a query optimization framework that we can exploit in the future, since performance is a major concern when designing decision-support applications that are integrally based on XML and XQuery.

We expressed in TAX the main usual OLAP operators: cube, rotate, switch, roll-up, drill-down, slice, dice, pull and push. By doing so, we significantly expanded the number of available XOLAP operators, since up to now, related papers only proposed at most three operators each (always including the cube operator). We have also implemented these XOLAP operators into a software prototype that helps generate the corresponding XQuery code. This querying interface is currently coupled to TIMBER XML-native DBMS, but it is actually independent and could operate onto any other DBMS supporting XQuery.

FUTURE TRENDS

Historically, XML data warehousing and OLAP approaches very often adapted existing, efficient relational solutions to the XML context. This was a sensible first step, but new approaches must now (and have definitely started to) take the specificities of XML data into account. More specifically, XML warehousing approaches now have to handle heterogeneous facts that may be described each by different dimension sets and/or hierarchy levels. Moreover, because of the variety of data sources, some fact and dimension data may also be missing. Finally, and especially when dealing with complex data such as multimedia data, measures may be non-numerical. For instance, if the observed fact is the health status of a patient, a radiography could be a measure.

Regarding XOLAP, let us take on an algorithmic metaphor. Most authors, and especially Wiwatwattana et al. (2007) and their excellent X^3 operator, have adopted a depth-first approach by fully developing one operator only, which is arguably of little use alone. On the other hand, we have adopted a breadth-first approach by proposing a wider range of operators that, however, only apply onto quite "regular" XML data. Both approaches should aim at completion, in breadth and depth, respectively, to achieve a full XOLAP environment; and are in our opinion complementary. In the next step of our work, we will indeed take inspiration from X^3's principle to enhance the rest of our XOLAP operators and truly make them XML-specific. For instance, we are actually currently working on performing roll-up

and drill-down operations onto the ragged hierarchies defined by Beyer et al. (2005).

CONCLUSION

More than a mere fashion related to XML's popularity, XML data warehousing and OLAP come from a true need. The complexity of data warehousing and OLAP technologies indeed makes them unattractive to many potential users and there is a growing need in the industry for simple, user-friendly Web-based interfaces, which is acknowledged by decision support system vendors (Lawton, 2006). Hence, though obvious and important difficulties do exist, including the maturity of XML-native DBMSs, especially regarding performance, we think that XML data warehousing and OLAP are promising approaches, and the best able to handle the numerous, so-called complex data (Darmont et al., 2005) that flourish on the Web, in a decision-support context.

REFERENCES

Agrawal, R., Gupta, A., & Sarawagi, S. (1997). Modeling Multidimensional Databases. In *13th International Conference on Data Engineering (ICDE 97), Birmingham, UK* (pp. 232-243). Los Alamitos: IEEE Computer Society.

Baril, X., & Bellahsène, Z. (2003). Designing and Managing an XML Warehouse. In Chaudhri, A. B., Rashid, A., & Zicari, R. (Eds.), *XML Data Management: Native XML and XML-enabled Database Systems* (pp. 455-473). Boston: Addison Wesley.

Beyer, K. S., Chamberlin, D. D., Colby, L. S., Özcan, F., Pirahesh, H., & Xu, Y. (2005). Extending XQuery for Analytics. In *2005 ACM SIGMOD International Conference on Management of Data (SIGMOD 05), Baltimore, USA* (pp. 503-514). New York: ACM Press.

Boukraa, D., Ben Messaoud, R., & Boussaïd, O. (2006). Proposition d'un Modèle physique pour les entrepôts XML. In *Atelier Systèmes Décisionnels (ASD 06), 9th Maghrebian Conference on Information Technologies (MCSEAI 06), Agadir, Morocco.* Agadir: MIPS-Maroc.

Boussaïd, O., Ben Messaoud, R., Choquet, R., & Anthoard, S., (2006). X-Warehousing: An XML-Based Approach for Warehousing Complex Data. In *10th East-European Conference on Advances in Databases and Information Systems (ADBIS 06), Thessaloniki, Greece* (pp. 39-54); *Lecture Notes in Computer Science*, 4152. Berlin: Springer.

Cheng, K., Kambayashi, Y., Lee, S. T., & Mohania, M. K. (2000). Functions of a Web Warehouse. In *Kyoto International Conference on Digital Libraries 2000* (pp. 372-379). Kyoto: Kyoto University.

Darmont, J., Boussaïd, O., Ralaivao, J. C., & Aouiche, K. (2005). An Architecture Framework for Complex Data Warehouses. In *7th International Conference on Enterprise Information Systems (ICEIS 05), Miami, USA* (pp. 370-373). Setúbal: INSTICC.

Golfarelli, M., Rizzi, S., & Vrdoljak, B. (2001). Data Warehouse Design from XML Sources. In *4th International Workshop on Data Warehousing and OLAP (DOLAP 01), Atlanta, USA* (pp. 40-47). New York: ACM Press.

Gyssens, M., & Lakshmanan, L. V. S. (1997). A Foundation for Multi-dimensional Databases. In *23rd International Conference on Very Large Data Bases (VLDB 97), Athens, Greece* (pp. 106-115). San Francisco: Morgan Kaufmann.

Hümmer, W., Bauer, A., & Harde, G. (2003). XCube: XML for data warehouses. In *6th International Workshop on Data Warehousing and OLAP (DOLAP 03), New Orleans, USA* (pp. 33-40). New York: ACM Press.

Jagadish, H. N., Lakshmanan, L. S. V., Srivastava, D., & Thompson, K. (2001). TAX: A Tree Algebra for XML. In *8th International Workshop on Database Programming Languages (DBPL 01), Frascati, Italy* (pp. 149-164). *Lecture Notes in Computer Science*, 2397. Berlin: Springer.

Jensen, M. R., Moller, T. H., & Pedersen, T. B. (2001). Specifying OLAP cubes on XML data. *Journal of Intelligent Information Systems*, 17(2-3), 255-280.

Lawton, G. (2006). Making Business Intelligence More Useful. *Computer*, 39(9), 14-16.

Niemi, T., Niinimäki, M., Nummenmaa, J., & Thanisch, P. (2002). Constructing an OLAP cube from distributed XML data. In *5th International Workshop on Data*

Warehousing and OLAP (DOLAP 02), McLean, USA (pp. 22-27). New York: ACM Press.

Nassis, V., Rajugan, R., Dillon, T. S., & Rahayu, J. W. (2005). Conceptual and Systematic Design Approach for XML Document Warehouses. *International Journal of Data Warehousing & Mining*, 1(3), 63-86.

Pedersen, D., Pedersen, J., & Pedersen, T. B. (2004). Integrating XML Data in the TARGIT OLAP System. In *20th International Conference on Data Engineering (ICDE 04), Boston, USA* (pp. 778-781). Los Alamitos: IEEE Computer Society.

Paparizos, S., Wu, Y., Lakshmanan, L. V. S., & Jagadish, H. V. (2004). Tree Logical Classes for Efficient Evaluation of XQuery. In *ACM SIGMOD International Conference on Management of Data (SIGMOD 04), Paris, France* (pp. 71-82). New York: ACM Press.

Pokorný, J. (2002). XML Data Warehouse: Modelling and Querying. In *5th International Baltic Conference (BalticDB&IS 02), Tallinn, Estonia* (pp. 267-280). Tallinn: Institute of Cybernetics.

Park, B. K., Han, H., & Song, I. Y. (2005). XML-OLAP: A Multidimensional Analysis Framework for XML Warehouses. In *7th International Conference on Data Warehousing and Knowledge Discovery (DaWaK 05), Copenhagen, Denmark* (pp. 32-42). *Lecture Notes in Computer Science*, 3589. Berlin: Springer.

Rajugan, R., Chang, E., & Dillon, T. S. (2005). Conceptual Design of an XML FACT Repository for Dispersed XML Document Warehouses and XML Marts. In *5th International Conference on Computer and Information Technology (CIT 05), Shanghai, China* (pp. 141-149). Los Alamitos: IEEE Computer Society.

Ravat, F., Teste, O., & Zurfluh, G. (2006). Constraint-Based Multi-Dimensional Databases. In Ma, Z. (Ed.), *Database Modeling for Industrial Data Management* (pp. 323-368). Hershey: Idea Group Publishing.

Rusu, L. I., Rahayu, J. W., & Taniar, D. (2005). A Methodology for Building XML Data Warehouses. *International Journal of Data Warehousing & Mining*, 1(2), 67-92.

Thomas, H., & Datta, A. (2001). A Conceptual Model and Algebra for On-Line Analytical Processing in Decision Support Databases. *Information Systems Research*, 12(1), 83-102.

Vrdoljak, B., Banek, M., & Rizzi, S. (2003). Designing Web Warehouses from XML Schemas. In *5th International Conference on Data Warehousing and Knowledge Discovery (DaWaK 03), Prague, Czech Republic* (pp 89-98); *Lecture Notes in Computer Science*, 2737. Berlin: Springer.

Wiwatwattana, N., Jagadish, H. V., Lakshmanan, L. V. S., & Srivastava, D. (2007). X^3: A Cube Operator for XML OLAP. In *23rd International Conference on Data Engineering (ICDE 07), Istanbul, Turkey* (pp. 916-925). Los Alamitos: IEEE Computer Society.

Wang, H., Li, J., He, Z., & Gao, H. (2005). OLAP for XML Data. In *5th International Conference on Computer and Information Technology (CIT 05), Shanghai, China* (pp. 233-237). Los Alamitos: IEEE Computer Society.

Xyleme, L., (2001). Xyleme: A Dynamic Warehouse for XML Data of the Web. In *International Database Engineering & Applications Symposium (IDEAS 01), Grenoble, France* (pp. 3-7). Los Alamitos: IEEE Computer Society.

Zhang, J., Wang, W., Liu, H., & Zhang, S. (2005). X-Warehouse: Building Query Pattern-Driven Data. In *14th international conference on World Wide Web (WWW 05), China, Japan* (pp. 896-897). New York: ACM Press.

KEY TERMS

Complex Data: Data that present several axes of complexity for analysis, e.g., data represented in various formats, diversely structured, from several sources, described through several points of view, and/or versioned.

Web Warehouse: "A shared information repository. A web warehouse acts as an information server that supports information gathering and provides value added services, such as transcoding, personalization." (Cheng et al., 2000)

XML Data Warehouse: A data warehouse managing multidimensionally modeled XML data.

XML Document Warehouse: An XML document repository dedicated to e-business and Web data analysis.

XML Graph: Data model representing the hierarchical nature of XML data. In an XML graph, nodes represent elements or attributes.

XML-OLAP or XOLAP: OLAP for XML; approaches and operators providing answers to analytical queries that are multidimensional in nature and applied to XML data.

XML-Enabled Association Analysis

Ling Feng
Tsinghua University, China

INTRODUCTION

The discovery of association rules from large amounts of structured or semi-structured data is an important data mining problem [Agrawal et al. 1993, Agrawal and Srikant 1994, Miyahara et al. 2001, Termier et al. 2002, Braga et al. 2002, Cong et al. 2002, Braga et al. 2003, Xiao et al. 2003, Maruyama and Uehara 2000, Wang and Liu 2000]. It has crucial applications in decision support and marketing strategy. The most prototypical application of association rules is market basket analysis using transaction databases from supermarkets. These databases contain sales transaction records, each of which details items bought by a customer in the transaction. Mining association rules is the process of discovering knowledge such as "*80% of customers who bought diapers also bought beer, and 35% of customers bought both diapers and beer*", which can be expressed as "*diaper* $\Rightarrow$ *beer*" (35%, 80%), where 80% is the *confidence* level of the rule, and 35% is the *support* level of the rule indicating how frequently the customers bought both diapers and beer. In general, an association rule takes the form $X \Rightarrow Y$ (s, c), where X and Y are sets of items, and s and c are support and confidence, respectively.

In the XML Era, mining association rules is confronted with more challenges than in the traditional well-structured world due to the inherent flexibilities of XML in both structure and semantics [Feng and Dillon 2005]. First, XML data has a more complex hierarchical structure than a database record. Second, elements in XML data have contextual positions, which thus carry the order notion. Third, XML data appears to be much bigger than traditional data. To address these challenges, the classic association rule mining framework originating with transactional databases needs to be re-examined.

BACKGROUND

In the literature, there exist techniques proposed to mine frequent patterns from complex tress and graphs databases. One of the most popular approaches is to use graph matching, which employs data structures like adjacency matrix [Inokuchi et al. 2000] or adjacency list [Kuramochi and Karypis 2001]. Another approach represents semi-structured tree-like structures using a string representation, which is more space efficient and relatively easy for manipulation [Zaki 2002]. This work concentrated on mining frequent tree structures within a forest, which can be extended to for mining frequent tree structures in XML documents. [Zhang et al. 2004, Zhang et al. 2005] proposed a framework, called XAR-Miner, which is directly applicable to mining association rules from XML data. Raw data in the XML document are preprocessed to transform to either an Indexed Content Tree (IX-tree) or Multi-relational databases (Multi-DB), depending on the size of XML document and memory constraint of the system, for efficient data selection and association rule mining. Task-relevant concepts are generalized to produce generalized meta-patterns, based on which the large association rules that meet the support and confidence levels are generated. Recently, Confronted with huge volume of XML data, [Tan, et al. 2005] proposed to generate candidates by model-validating, so that there is no time wasted in deriving invalid candidates which will be discarded at later stages. The algorithm processes an XML document directly taking into account the values of the nodes present in the XML tree, so the frequent item-sets generated contain both node names and values in comparison to the TreeMiner approach, which only generates frequent tree structures. The experiments with both synthetic and real life data sets demonstrate the efficiency of this approach.

MAIN FOCUS

The Framework

Under the traditional association rule framework, the basic unit of data to look at is database *record*, and the construct unit of a discovered association rule is *item* which has an *atomic value*. These lead us to the following two questions: 1) *what is the counterpart of record* and 2) *what is the counterpart of item* in mining association relationships from XML data? [Feng et al. 2003, Feng and Dillon 2004]. This investigation focuses on rule detection from a collection of XML documents, which describe the same type of information (e.g., customer order, etc.). Hence, each of XML documents corresponds to a database record, and possesses a tree-like structure. Accordingly, we extend the notion of associated item to an XML fragment (i.e., tree), and build up associations among trees rather than simple-structured items of atomic values. For consistency, we call each such kind of trees a **tree-structured item** to distinguish it from the traditional counterpart item. With the above extended notions, we propose an **XML-enabled association rule framework**. From both structural and semantic aspects, XML-enabled association rules are more powerful and flexible than the traditional ones.

Definition 1 *Let T denote a set of trees (tree-structured items). An* **XML-enabled association rule** *is an implication of the form $X \Rightarrow Y$, which satisfies the following two conditions:*

1. $X \subset T$, $Y \subset T$, *and* $X \cap Y = \phi$;
2. *for* $\forall T, T' \in (X \cup Y)$, *there exists no such tree* T''' *where* T'' *is a subtree of* T *and* T'.

Different from classical association rules where associated items are usually denoted using simple structured data from the domains of basic data types, the items in XML-enabled association rules can have a hierarchical tree structure, as indicated by the first clause of the definition. Here, it is worth pointing out that when each of the tree-structured items contains only one basic root node, the XML-enabled association rules will degrade to the traditional association rules. The second clause of the definition requires that in an XML-enabled association rule, no common sub-trees exist within any two item trees in order to avoid redundant expression.

Figure 1 illustrates some XML-enabled association rule examples. Thanks to XML, XML-enabled association rules are more powerful than traditional association rules in capturing and describing association relationships. Such enhanced capabilities can be reflected from both a structural as well as a semantic point of view:

- Association items have hierarchical tree structures, which are more natural, informative and understandable (e.g., Rule 1 & 2 in Figure 1).
- Associated items inherently carry the *order* notion, enabling a uniform description of association and sequence patterns within one mining framework (e.g., Rule 1 states the sequence of books to be ordered, i.e., "*Star War I*" proceeding "*Star War II* " on a customer's order).
- Associated items can further be constrained by their context positions, hierarchical levels, and weak/strong adhesion in the corresponding XML data to be mined. (e.g., Rule 1 indicates the contextual appearances of BOOKs on the order).
- Association relationships among structures and structured-values can also be captured and described (e.g., Rule 2 states that a student orders some flowers from a shop, and leaves detailed content of FLOWER element such as the kind of flowers and quantity, etc. aside).
- Auxiliary information which states the occurrence context of association relationships can be uniformly self-described in the mining framework (e.g., Rule 1 indicates that only male people have such as order pattern).

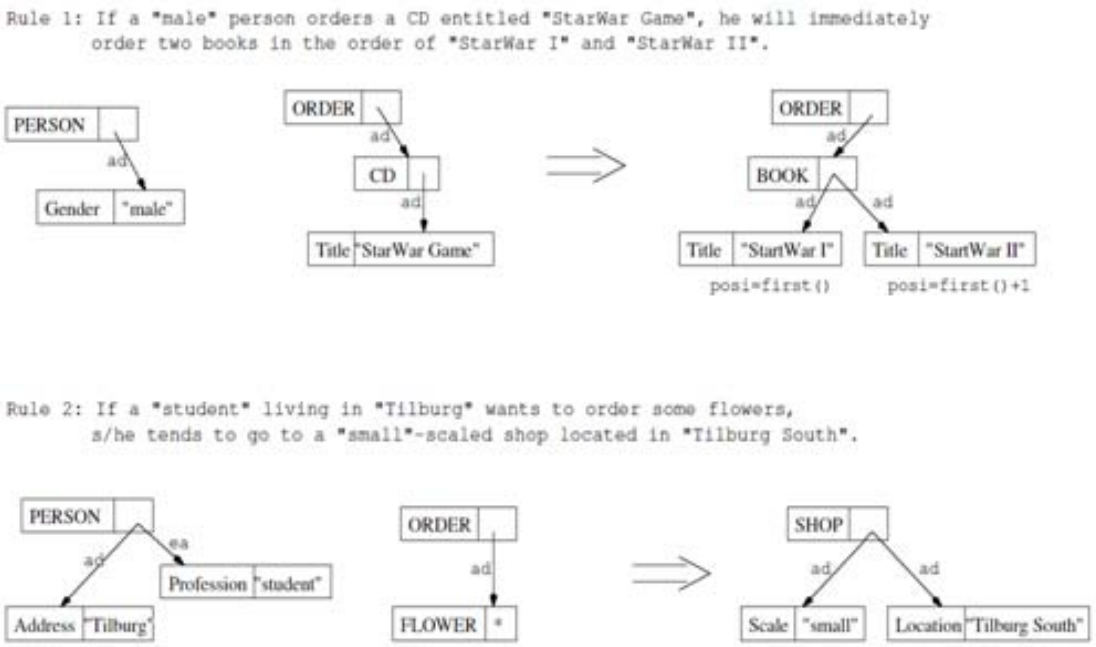
Similar to traditional association rules, we use support and confidence as two major measurements for XML-enabled association rules.

Definition 2 *Let D be a set of XML documents. The* **support** *and* **confidence** *of an XML-enabled association rule $X \Rightarrow Y$ are defined as follows:*

$$\text{support}(X \Rightarrow Y) = \frac{|D_{xy}|}{|\mathrm{D}|}, \quad \text{confidence}(X \Rightarrow Y) = \frac{|D_{xy}|}{|D_x|}$$

where $D_{xy} = \{doc | \forall T \in (X \cup Y)(T \in_{tree} doc)\}$, and $D_x = \{doc | \forall T \in X(T \in_{tree} doc)\}$.

Figure 1. XML-enabled association rule examples



The merit of such a support and confidence measurement definition in the context of XML association mining is that it sticks to the main stream of association analysis, where the counter part of "atomic item" concept in the classical association analysis is "tree-structured item", and that of the classical "transaction" concept corresponds to an XML document. This could help extend traditional methods and facilitate the development of new techniques in the newly emerging XML domain.

Template-Guided Mining of XML-Enabled Association Rules

The template-guided mining of XML-enabled association rules proceeds in three steps.

Phase-1: Transforming Tree-Structured Data into Sequences

To prepare for efficient mining, our first step is to transform each tree in the XML database and each tree variable in the template expression into a sequence while preserving their hierarchical structures. We employ the encoding technique, recently developed by Wang et al. for efficient indexing and querying XML data [Wang et al. 2003], to do such transformation. That is, a tree T is transformed into a sequence *Transform(T)* = $\langle(a_1, p_1), (a_2, p_2), \ldots, (a_n p_n)\rangle$, where a_i represents a node in the tree T, p_i is the path from the root node to node a_i, and $a_1, a_2, \ldots,$ an is the preorder traversal of the tree [Wang et al. 2003].

Phase-2: Discovering Structurally Containing Relationships by Sub-sequence Matching

With the structure-encoded sequences, checking the embedding relationship (i.e., whether an XML document contains a template tree variable) degrades to non-contiguous subsequence matching.

Phase-3: Correlating Concrete Contents with Structures

For each document obtained after Phase-2, the last step is to instantiate every unknown symbol (either node name or node value) in the template with a concrete content extracted from the document, which also observes content constraint(s) as indicated by the template.

To demonstrate the applicability of the XML-enabled association framework, we performed constraint-based XML-enabled association rule mining on the real-life DBLP data (http://dblp.uni-trier.de/xml/), which contains a huge number of bibliography entries in XML. Suppose we are interested in finding out who publishes frequently together with *Michael J. Franklin* in conference proceedings. Figure 2 is the template we use while mining the DBLP data. To restrict the mining space, we pre-process the original big DBLP file of size 187 MB by running an XQuery to extract in total 284663 fragments enclosed by <inproceedings>.....</inproceedings>. After discovering 52 XML segments which structurally contain the two tree variables of the template, i.e., conference papers written by *Michael J. Franklin*, we hash the corresponding co-authors' names into a hash table, and obtain the following rules when the support threshold is set to 0.002%.

Here are some rules, where author is

"*Stanley B. Zdonik*" (*support* = $\frac{10}{284663}$ = 0.004%, *confidences* = $\frac{10}{52}$ = 19.2%)

"*Michael J. Franklin*" (*support* = $\frac{7}{284663}$ = 0.003%, *confidences* = $\frac{7}{52}$ = 13.5%)

"*Samuel Madden*" (*support* = $\frac{6}{284663}$ = 0.003%, *confidences* = $\frac{6}{52}$ = 11.5%)

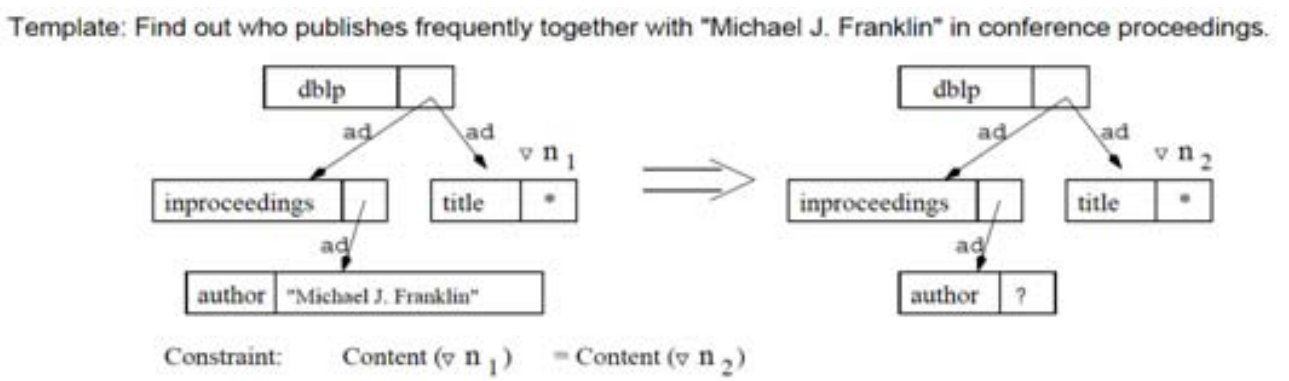
FUTURE TRENDS

From a structural as well as a semantic point of view, XML-enabled association rule framework is more flexible and powerful than the traditional one in representing both simple and complex structured association relationships. However, mining XML-enabled association relationships poses more challenges for efficient processing than mining traditional association rules. To address the trade-off between the enhanced capabilities on the one side and mining performance on the other side, an appropriate template model will play an necessary and important role in XML-enabled association mining.

CONCLUSION

We integrate the newly emerging XML technology into association rule mining technique. An extended XML-enabled association rule framework is presented, with the aim to discover associations inherent in massive amounts of XML data. From a structural as well as a semantic point of view, such a framework is more flexible and powerful than the traditional one in representing both simple and complex structured association relationships. Experimental results on the real-life DBLP data are presented to demonstrate the applicability of the framework.

Figure 2. The mining constraint for the experiment on DBPL data



REFERENCES

Agrawal, R., Imielinski, T., and Swami, A. (1993). Mining associations between sets of items in massive databases, *Proc. of the ACM SIGMOD International Conference on Management of Data*, pages 207-216.

Agrawal, R. and Srikant, R. (1994). Fast algorithms for mining association rules, *Proc. of the 20th Conference on Very Large Data Bases*, pages 478-499.

Maruyama, K. and Uehara, K. (2000). Mining association rules from semi-structured data. *Proc. of the ICDCS Workshop of Knowledge Discovery and Data Mining in the World-Wide Web*, Taiwan.

Wang, K. and Liu, H. (2000). Discovering structural association of semistructured data. *IEEE Transactions on Knowledge and Data Engineering*, 12(2), pages 353-371.

Inokuchi, A., Washio, T., and Motoda, H. (2000) An apriori-based algorithm for mining frequent substructures from graph data. *Proc. of the 4th European Conference on Principles and Practice of Data Mining and Knowledge Discovery*, pages 13-23.

Kuramochi, M. and Karypis, G. (2001) Frequent Subgraph Discovery. *Proc. of the IEEE International Conference on Data Mining*, pages 313–320.

Braga, D., Campi, S., Ceri, S., Klemettinen, M., and Lanzi, P. (2003). Discovering Interesting Information in XML Data With Association Rules. *Proc. of the 18th Symposium on Applied Computing*.

Braga, D., Campi, S., Klemettinen, M., and Lanzi, P. (2002). Mining Association Rules From XML Data. *Proc. of the 4th Intl. Conf. on Data Warehousing and Knowledge Discovery*.

Cong, G., Yi, L., Liu, B., and Wang, K. (2002). Discovering Frequent Substructures from Hierarchical Semi-structured Data, *Proc. of SIAM DM'02*.

Zaki, M. J. (2002). Efficient Mining of Trees in the Forest. *Proc. of the* Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 71-80.

Dunham Margaret H. (2003). Data Mining: Introductory and Advanced Topics. Prentice Hall.

Feng, L., Dillon, T., Weigand, H., and Chang, E. (2003). An xml-enabled association rule framework. *Proc. of the 14th Intl. Conf. on Database and Expert Systems Applications*, Prague, Czech Republic, pages 88-97.

Wang, H., Park, S., Fan, W., and Yu, P. (2003). ViST: A dynamic index method for querying XML data by tree structures. *Proc. of the ACM SIGMOD Intl. Conf. on Management of Data*, California, USA, pages 110–121.

Zhang, J., Ling, T. W., Bruckner, R.M., Tjoa, A.M., and Liu, H. (2004). On Efficient and Effective Association Rule Mining from XML Data. *Proc. of the 15th Intl. Conf. on Database and Expert Systems Applications*, Zaragosa, Spain, pages 497-507.

Feng, L. and Dillon, T. (2004). Mining Interesting XML-Enabled Association Rules with Templates. *Proc. of the Third International Workshop on Knowledge Discovery in Inductive Databases*, Italy, September 2004, page 61-72, also in the Lecture Notes in Computer Science LNCS 3377, ISBN: 3-540-25082-4, 2005, page 66-88.

Feng, L. and Dillon, T. (2005). An XML-Enabled Data Mining Query Language XML-DMQL (invited paper). *International Journal of Business Intelligence and Data Mining*, ISSN: 1743-8187, Vol. 1, No. 1, page 22-41.

Zhang, S., Zhang, J., Liu, H., Wang, W. (2005). XAR-Miner: Efficient Association Rules Mining for XML Data. *Proc. of the Intl. Conf. on WWW*, Chiba, Japan, pages 894-895.

Tan, H., Dillon, T., Feng, L., Chang, E., and Hadzic, F. (2005). *X3-Miner: Mining Patterns from XML Database*, *Proc. of International Conference on Data Mining*, Skiathos, Greece, page 287-299.

KEY TERMS

Association: An association indicates the correlation relationship among items within the same transactions.

Mining Template: A mining template is an expression used to constrain the mining process.

Support and Confidence: Support and confidence are two measures used to describe the statistic significance of association rules.

Tree: A tree implies a tree data structure which has one root node, and any child node has only one parent node.

Tree-Structured Item: A tree-structured item indicates the item which has a tree structure.

XML: XML stands for eXtensible Markup Language.

XML-Enabled Association: XML-enabled association indicates the correlation relationship among tree-structured items which exists within the same XML data.

Index

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Z